NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

PROCEDURAL GUIDE FOR MODELING AND ANALYZING THE FLIGHT DYNAMICS OF THE SH-60B HELICOPTER USING FLIGHTLAB

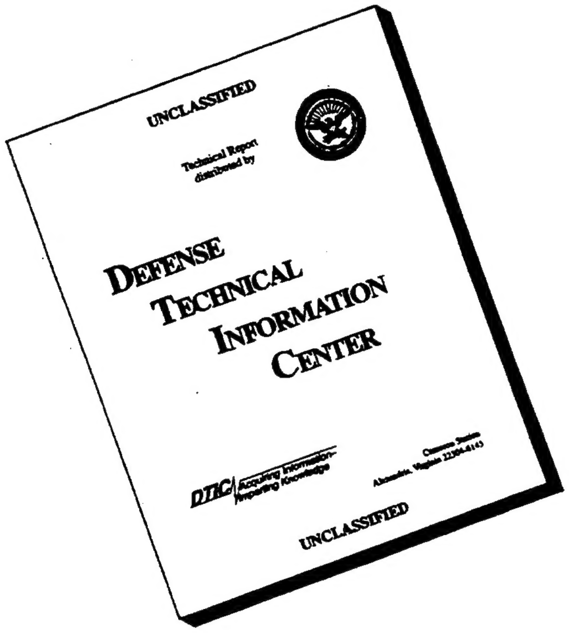
by

Roy C. Wagner

September, 1995

Thesis Advisor: E. R. Wood

Approved for public release; distribution is unlimited.


UNCLASSIFIED
Technical Report distributed by
DEFENSE TECHNICAL INFORMATION CENTER
DTIC Acquiring Information- Imparting Knowledge
Cameron Station Alexandria, Virginia 22304-6145
UNCLASSIFIED

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 1995	3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE PROCEDURAL GUIDE FOR MODELING AND ANALYZING THE FLIGHT DYNAMICS OF THE SH-60B HELICOPTER USING FLIGHTLAB		5. FUNDING NUMBERS
6. AUTHOR(S) Roy C. Wagner		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.		
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

This thesis uses Flightlab to model and analyze the flight dynamics of the SH-60B Seahawk helicopter. Flightlab runs on computers utilizing the UNIX™ operating system and is used for design, analysis and testing of an aircraft using non-linear modeling techniques. It is capable of modeling conventional main rotor-tail rotor and tandem rotor helicopters and tilt rotor aircraft. A procedural guide for modeling and analyzing a single main rotor helicopter is presented using the SH-60B. The non-linear response from lateral and longitudinal cyclic, main rotor collective and tail rotor collective inputs are presented. Flightlab is also capable of reducing the non-linear model to a linear model. The linear and non-linear models are then compared. The purpose of this thesis is to present a guide for using Flightlab to model and analyze an existing helicopter design, and also to have in place a well tested model to be used for further research.

14. SUBJECT TERMS Helicopter flight modeling, analysis, and design. Flight simulation. Sikorsky SH-60B helicopter			15. NUMBER OF PAGES 179
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

PROCEDURAL GUIDE FOR MODELING AND ANALYZING THE FLIGHT DYNAMICS OF THE SH-60B HELICOPTER USING FLIGHTLAB

Roy C. Wagner
Lieutenant, United States Navy
B.S., Pennsylvania State University, 1984

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1995

Author: Roy C Wagner
Roy C. Wagner

Approved by: E. R. Wood
E. R. Wood, Thesis Advisor

Hossein-ali Saberi
Hossein-Ali Saberi, Second Reader

Daniel J Collins
D. J. Collins, Chairman
Department of Aeronautics and Astronautics

ABSTRACT

This thesis uses Flightlab to model and analyze the flight dynamics of the SH-60B Seahawk helicopter. Flightlab runs on computers utilizing the UNIX^{TM} operating system and is used for design, analysis and testing of an aircraft using non-linear modeling techniques. It is capable of modeling conventional main rotor-tail rotor and tandem rotor helicopters and tilt rotor aircraft. A procedural guide for modeling and analyzing a single main rotor helicopter is presented using the SH-60B. The non-linear response from lateral and longitudinal cyclic, main rotor collective and tail rotor collective inputs are presented. Flightlab is also capable of reducing the non-linear model to a linear model. The linear and non-linear models are then compared. The purpose of this thesis is to present a guide for using Flightlab to model and analyze an existing helicopter design, and also to have in place a well tested model to be used for further research.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to several people who provided me with invaluable assistance in completing this thesis. Dr. Keeyoung Choi, from Advanced Rotorcraft Technologies, helped me in learning, understanding, and building the SH-60B model in Flightlab. Mr. Tom Lawrence, from Sikorsky Aircraft, was my point of contact for the SH-60B data and assisted me with explanations of the control system. I would also like to thank my thesis advisor from the Naval Postgraduate School, Dr. E.R. Wood, and my second reader, Dr. Hossein-Ali Saberi, from Advanced Rotorcraft Technologies for their time and expertise.

I. INTRODUCTION

A. BACKGROUND

The advent of high speed processors and realistic computer generated dynamic displays have made the computer an indispensable tool for rotorcraft design and analysis. The hardware though is only one part of the equation, the other being software. The four major helicopter manufacturers in the United States, McDonnell Douglas, Sikorsky Aircraft, Bell Helicopter Textron and Boeing Helicopters have all invested heavily in computer flight simulation, modeling, and engineering analysis. The most ambitious is probably Sikorsky Aircraft's expansion of its simulation facility to 300,000 square feet, mainly to support the LHX program [Ref. 1: p. 153]. The selection of the Boeing/Sikorsky team to build the new Light Helicopter (LHX) for The U. S. Army was the result of competition based solely on the flight simulation of an engineering design [Ref. 2: p. 1]. This was accomplished prior to the first prototype being built. High end rotorcraft modeling software is used across all lines of analysis and design, not just flight modeling. It is also used for hardware in the loop testing, piloted simulations and rapid prototyping of new system designs.

The large interest in computer flight analysis and simulation has led to the development of a software package called Flightlab. This program provides greater fidelity in modeling rotorcraft over purely mathematical based models usually used in MATLAB™ or other computational software. Flightlab's strength in this area is due to the fact that rotorcraft models are constructed using physical components, i.e. a hinge, wing or distributed mass to mention a few. The computations to handle the reaction of aerodynamic and gravitational forces between the model components is handled by the software. The user does not have to derive a mathematical expression to represent the dynamics. This approach leads to a nonlinear analysis which is valid over the entire flight envelope as opposed to a linear model which is valid only within the small linear region.

A second benefit is that the rotor system degrees of freedom are calculated in addition to the six degrees of freedom of the body. This permits a more detailed analysis to be conducted on the model and thus an increase in the accuracy of the results. Flightlab is also capable of real time flight simulation for piloted evaluations. This requires the use of parallel processors.

B. OVERVIEW OF FLIGHTLAB

The Flightlab program, current version 2.5, is written in a higher level language which uses a combination of FORTRAN and C computer languages. This fact is transparent to the user, but a familiarity with the UNIX™ operating system is required.

There are three programs within Flightlab for nonlinear modeling of aircraft, Model Editor, Gscope, and Model Analysis. The Model Editor is a high level modeling tool that allows for quick construction of a model through the use of standardized model structures. Each of the eight standardized model structures, also called supercomponents, facilitate for a modular design which allow for easy changes in model design. In addition, the Model Editor provides for selective fidelity modeling options. For example, the rotor system can be modeled as either a rigid or elastic system. Pertinent data for selected supercomponents is entered into data templates either directly as numeric data or the names of data files for more involved data, for example coefficient of lift versus angle of attack of the fuselage.

Gscope provides an environment similar to SIMULINK™ for constructing systems or components of the aircraft. Once a system is constructed in Gscope, it is linked to the model created in the Model Editor. Gscope is most often used when a particular system of the aircraft under study differs from the generalized model structure present in the Model Editor. It is even possible to construct an entire model in Gscope without using Model Editor, although with a much higher workload.

The Model Analysis program contains numerous scripts accessed in a menu format for analyzing the dynamics of the model. It is a very extensive analysis program and can also be modified to meet the needs of the user. The second aspect of the Model Analysis is a command line interface with many of the basic features of MATLAB™ for matrix manipulation and plotting of data. Flightlab uses the interactive plotting capability of XMGR.

C. MOTIVATION

The Flightlab program provides for dynamic real time simulation of an aircraft model that can be used in a design course within a one semester time frame, but the rapid improvements in the program have outpaced the development of user manuals provided by the company. The developers of Flightlab provide five user manuals. Scope Users Tutorial [Ref. 3]; Component Reference Manual [Ref. 4]; Scope Theory Manual [Ref. 5] ; Scope Command Reference Manual [Ref. 6]; and the latest, Flightlab Training Manual [Ref. 7].

The first four manuals were written for one of the first versions of Flightlab, which only had Gscope as its modeling and analysis tool. It required a great deal of time and expertise to implement and analyze a design and thus would not allow for the design to be completed in one semester unless the user was already well versed in its use. In addition, a previous Naval Postgraduate School thesis, Procedural Guide For Modeling and Analyzing the Flight Characteristics of a Helicopter Design Using Flightlab [Ref. 2], covered the use of Gscope for the complete modeling of an aircraft, where references 3 thru 6 covered much of the basics of modeling various items, but they do not show how to model a complete aircraft.

Flightlab version 2.4 introduced two additional tools, Model Editor and Model Analysis. Reference 7 is a tutorial on how to use these two tools. This improvement drastically reduced the amount of time needed to implement and begin testing a model. The model editor contains generalized aircraft components of varying degrees of fidelity, which are chosen based on simulation requirements. Data is then entered in a graphical user interface format, the file saved and analysis begun. Flightlab version 2.5, as of now, has no manuals describing it's improvements over previous versions.

The purpose of this thesis is two fold. First, to provide a structured, logically sequenced instruction manual for the implementation and testing of an existing aircraft design. Also, to explain some of the finer points of the program that are usually experienced only after using such a program.

Second, to model and analyze the flight dynamics of the SH-60B Seahawk helicopter and compare the results with those obtained by Sikorsky Aircraft using their analysis program, GenHel. Aerodynamic and control system data for the SH-60B was supplied by Sikorsky Aircraft. The flight characteristics of the closed and open loop model will also be studied.

II. FLIGHTLAB TUTORIAL

A. GETTING STARTED WITH FLIGHTLAB

Flightlab is currently installed on the Naval Postgraduate School Department of Aeronautics SGI Indigo II workstation hawkeye, but can be easily accessed by other aeronautics workstations utilizing the UNIX™ operating system environment. The procedure by which will be explained shortly. Setting up an account to run Flightlab requires modifying the .login and .cshrc files as in Table I.

```
.login file
    setenv FL_DIR /usr/local/flightlab-2.5
    source $FL_DIR/SOURCE.ME
    setenv PS_PRINTER 'hp3si_1'
    setenv FL_TERM xterm

.cshrc file
    set path=( /usr/local/flightlab-2.5 /usr/local/
            flightlab-2.5/bin $path)
    xhost hawkeye.aa.nps.navy.mil
```

Table I. Login Files Modification

After modifying the .login and .cshrc files, the process of starting Flightlab will vary slightly depending on whether the user is physically located at hawkeye or at another workstation. If at hawkeye, the program is started by typing "flightscope &" at the users home directory. The ampersand is optional, but allows the shell window to be used while the program is running in the background. This is convenient as it allows for editing of files and viewing the various Flightlab windows at the same time. If at a workstation other than hawkeye, follow the procedures in Table II.

```
rlogin hawkeye.aa.nps.navy.mil
    enter user name and password
setenv DISPLAY phantom.aa.nps.navy.mil:0.0 *
flightscope &
```

*Enter the terminal name as appropriate, phantom is used as and example. Ensure that the :0.0 is included immediately following the terminal name.

Table II. Remote Login Procedures

Flightlab will begin by displaying the main window as shown in Figure 1. The three main tools of the program, Model Editor, Gscope, and Model Analysis are each accessed by clicking once on the appropriate square with the left mouse button.

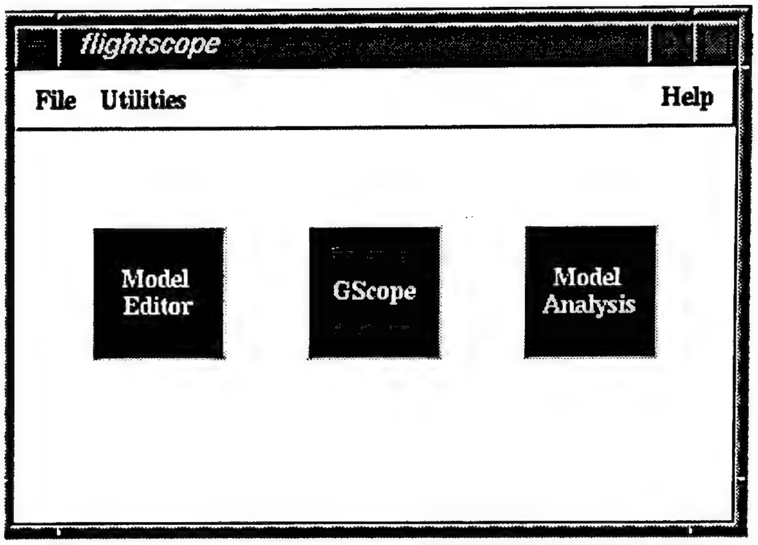


Figure 1. Flightlab Main Window

B. MODEL EDITOR

The Model Editor is where a majority of the data for the aircraft under study will be entered. When selected, it brings up a model tree which contains all model structures necessary for the complete modeling of an aircraft. Figure 2 is a compressed version of the full model tree, but does show the eight supercomponents and a number of selected subcomponents. The eight supercomponents are, EnvironAtmos, Rotor1, Rotor2, Wing, Propulsion, Airframe, Flight Control, and Aero Interference.

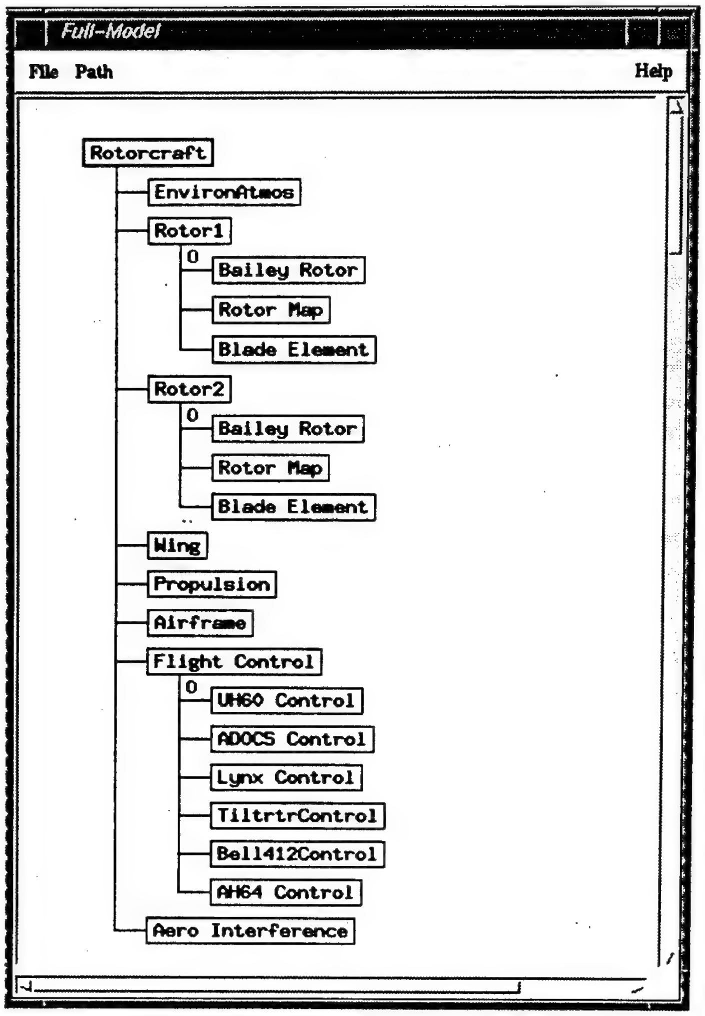
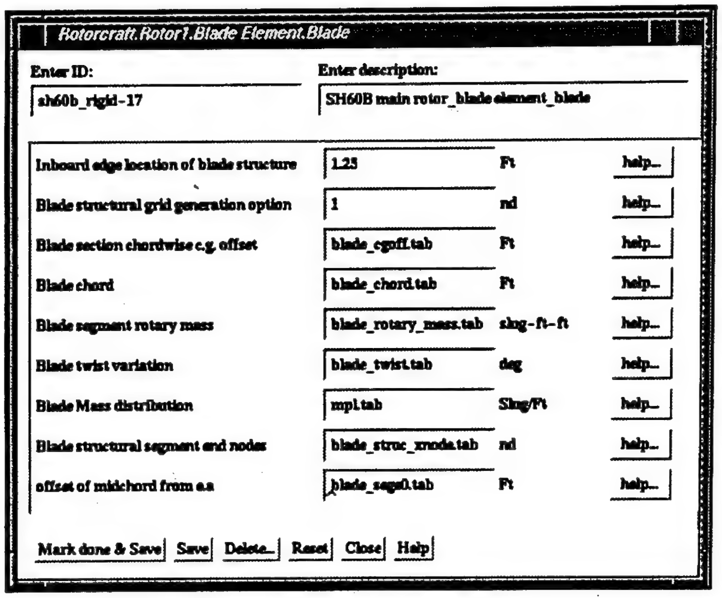


Figure 2. Model Editor Tree

Construction of a model begins by first selecting the Rotorcraft option at the top of the tree with one click of the left mouse button. The Rotorcraft option will then become enclosed in a red box indicating that it has been selected and will become part of the model. This is true for all supercomponents and subcomponents selected.

Next, the appropriate supercomponents and subcomponents are selected in the same fashion. Flightlab will not allow a subcomponent to be selected prior to it's corresponding supercomponent being selected and enclosed in a red box. This is what is refereed to as a top down design. Hereafter supercomponents and subcomponents will be generically refered to as components.

After all desired components are selected, a double click with the left mouse button on any of the previously selected components will bring up it's corresponding data template, although not all components have a data template. A representative data template is shown in Figure 3.



Enter ID: sh60b_rigid-17

Enter description: SH60B main rotor_blade element_blade

Inboard edge location of blade structure	1.25	Ft	help...
Blade structural grid generation option	1	nd	help...
Blade section chordwise c.g. offset	blade_cgoff.tab	Ft	help...
Blade chord	blade_chord.tab	Ft	help...
Blade segment rotary mass	blade_rotary_mass.tab	slug-ft-ft	help...
Blade twist variation	blade_twist.tab	deg	help...
Blade Mass distribution	mpl.tab	Slug/Ft	help...
Blade structural segment end nodes	blade_struc_xnode.tab	nd	help...
offset of midchord from e.a	blade_sega0.tab	Ft	help...

Figure 3. Data Template

It is important to note that not all of the eight supercomponents must be selected in order to start analyzing the model. Flightlab is flexible enough to allow each system to be constructed in sequence and then tested to allow for easy debugging both in syntax and logic implemented. For example, the EnvironAtmos and Rotor 1 components can be selected, the appropriate data entered, and then the model analyzed as an isolated rotor in the free stream.

Figure 3 shows that the required data is entered either as numerics or the name of a file which contains the necessary data in table form. The directory path for data files is specified in the "Path" menu on the upper left corner of the model tree. Beneath "Path" select the "Path for Tables..." option. Flightlab will respond with a window to enter the desired path. All data files refereed to in the model must be present in this path.

Data table files are called by one of two methods. File names with the .sav suffix are read in with the "read" command and stored in RAM to the variable assigned. For example, the table corresponding to blade_cg.tab in Figure 3 would be: bchord=read("directory path/blade_cg.tab "). Comments in .sav files are delineated by the pound sign, #, directly preceding the comment.

The second method corresponds to files with the .tab suffix. Here the "load" command is used and the variable name is included in the data file as the first readable element. The syntax for the load command is: load("directory path/cl1.sav"). Comments in .tab files are delineated by the pound sign , #, directly preceding the comment.

The two methods for calling a data file also differ in the way the data is read from the file and entered into the matrix. The load command requires that the variable name for the data file be the first readable element in the file, immediately followed by three numbers which specify the size matrix the data will be entered into. The first is the number of rows, the second the number of columns and the third the depth of the table, usually zero. The data file will be read from left to right top to bottom filling in the matrix columns first, i.e. first row first column, second row first column to the end of the column and then starts on the second column. Table III demonstrates how to enter the data of a two dimensional array into a data file to be called by the load command.

The read command only requires two numbers to specify the matrix size and does not require the variable name in the data file, as it is assigned by the calling command. The data is read from the file left to right top to bottom as before, but the data is entered into the matrix in a different fashion. The rows are filled in first followed by the columns, i.e., first row first column, first row second column to the end of the first row and then starts on the second row. Table IV demonstrates how to enter the data blade chord vs blade station to be called by the read command.

After all the data is entered, click on the "mark save and done" and then "close" boxes on the bottom of the data template. The template will close and its corresponding component on the model tree will now be enclosed in a red box with the left half colored yellow. This indicates that data entry is complete at this node. If a data file name entered on the data template is not found in the directory path specified, Flightlab will give an alert stating so, but the alert can be overridden and the node saved regardless.

Data given in following format

input 1 = sideslip angle -10 to 10 delta 5
input 2 = angle of attack -10 to 10 delta 10
output = pitching moment

```
                              angle of attack = -10
        20.0 18.0 0.0 0.0 -7.0
                               angle of attack = 0
      -39.0 -5.0 0.0 -16.0 -55.0
                              angle of attack = +10
      -56.0 -15.0 75.0 128.0 171.0
```

Data file format

```
variable name 5 3 0

   20.0    18.0    0.0    0.0    -7.0
  -39.0    -5.0    0.0  -16.0   -55.0
  -56.0   -15.0   75.0  128.0   171.0
```

If the variable name is called up in the GScope command line, the matrix will be displayed as follows:

```
20.0   -39.0   -56.0
18.0    -5.0   -15.0
 0.0     0.0    75.0
 0.0   -16.0   128.0
-7.0   -55.0   171.0
```

Table III. Data File Structure for Load Command

Data given in following format

Blade station	.041	.046	.100	.200	.300	.400	.500	.600	.700	.800	.900	1.00
Chord	0.00	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73

Data file format

```
12 2
  0.0410   0.00     0.0460   1.73   0.1000   1.73
  0.2000   1.73     0.3000   1.73   0.4000   1.73
  0.5000   1.73     0.6000   1.73   0.7000   1.73
  0.8000   1.73     0.9000   1.73   1.0000   1.73
```

If the variable name is called up in the Gscope command line, the matrix will be display as follows:

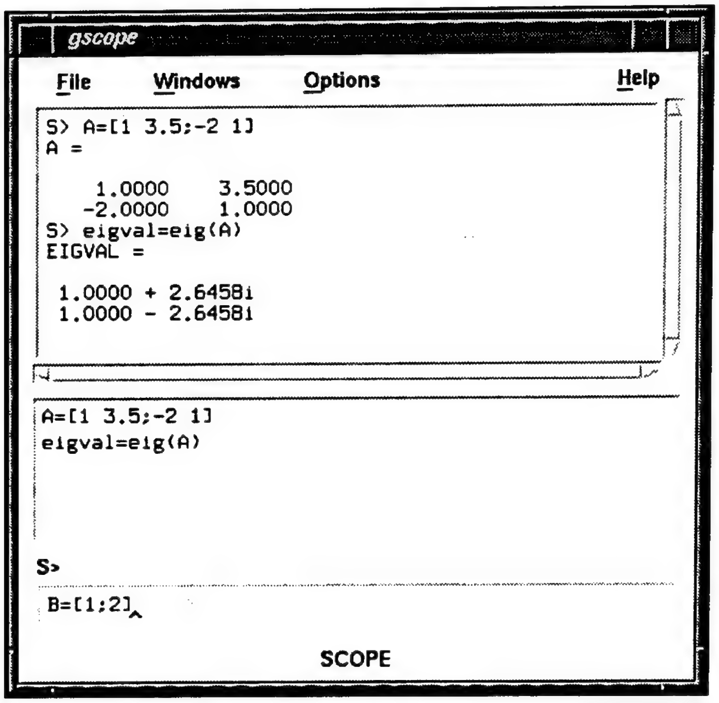
```
0.4100   0.00
0.0460   1.73
0.1000   1.73
0.2000   1.73
0.3000   1.73
0.4000   1.73
0.5000   1.73
0.6000   1.73
0.7000   1.73
0.8000   1.73
0.9000   1.73
1.0000   1.73
```

Table IV. Data File Structure for Read Command

The model can now be saved by accessing the "File" menu at the upper left corner of the model tree and selecting the "Save model as..." option. Prior to saving the model though the "Path for Local Database...", beneath the "Path" option must be set. Flightlab comes with a selection of completed ready to run models that are located in a central database. The path option for these are: /usr/local/flightlab-2.5/flightscope/models/uh60. Generally, models to be saved in the users account will have the following path: /d4/user name/flightscope/models/rotorcraft.

C. GSCOPE

When selected, the Gscope window will be displayed as shown in Figure 4. The purpose of Gscope is to provide a tool to build dynamic models, enter associated data and script files, and conduct analysis on the model using the Scope language. Scope consists of a set of commands and functions that support linear systems analysis, parameter identification, eigenvalue analysis, fourier transforms and general matrix manipulation [Ref. 1: p. 22]. Scope Command Reference, Reference 4, gives the basic operations, syntax, functions, and commands supported by Scope. These basic commands can be combined to form new functions, script files, that can be edited and then executed. These user defined script files can be as simple as a series of repetitive calculations to executing an entire rotorcraft model and plotting the desired output.



```
S> A=[1 3.5;-2 1]
A =

    1.0000     3.5000
   -2.0000     1.0000
S> eigval=eig(A)
EIGVAL =

 1.0000 + 2.6458i
 1.0000 - 2.6458i
```

```
A=[1 3.5;-2 1]
eigval=eig(A)

S>
B=[1;2]
```

Figure 4. Gscope Main Window

Gscope has five window options, main window, model window, editor window, help window, and plot window. All are accessed via the "Windows" pull down menu at the top of the main window.

1. Main Window

Referring to Figure 4, the main window has three basic areas, the display, history and command entry screens. The display screen is the upper area of the main window below the pull down menus. This is where the commands and output of the Gscope program are displayed.

The middle area is the command history screen. It is a buffer which holds the twenty most previous commands entered. Previous commands may be copied to the current command line by clicking on the desired one with the left mouse button. After clicking on a command, the others can be scrolled through by using the up and down arrow keys. The command line screen is just below the S> at the bottom of the main window. Single line entry of Scope commands are entered here.

2. Model Window

The model window can be accessed by selecting it beneath the "Windows" pull down menu at the top of the main window. The model window shown in Figure 5 is used to build and design the model.

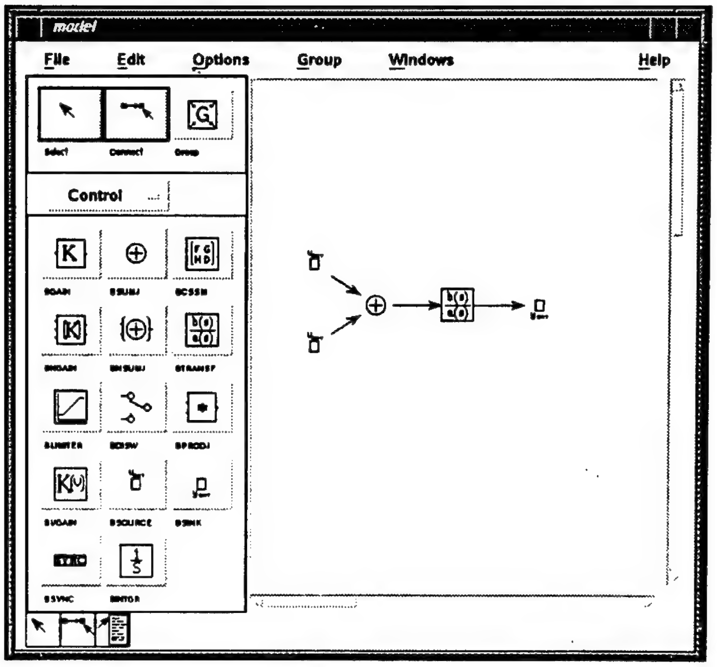
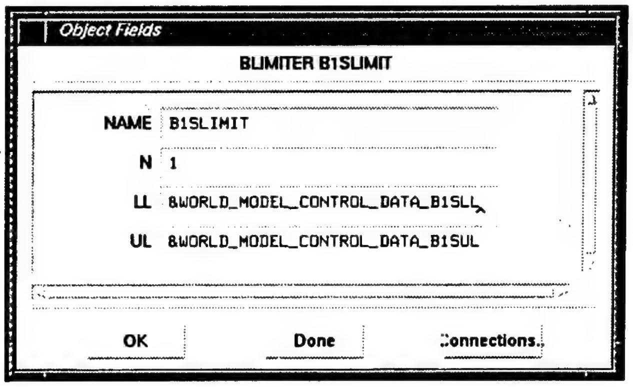


Figure 5. Model Window

Construction of a model is accomplished by selecting and placing components in the model screen, connecting them and then entering the pertinent data in dialog boxes. Components were designed to model a specific physical entity such as a hinge, a distributed mass, or a wing, to mention just a few. For convenience, components are grouped into classes. The five classes are kinematic, aero, control, analytic and solution. The icons of the components within the five classes are included in Appendix A. Each component class is accessed on the left side of the model window. Figure 5 shows the control class. To display a different class, place the cursor over the class name and press the left mouse button, a menu of the five classes will be displayed. Simply move the cursor to the desired one and release the mouse button.

Components are selected by first placing the cursor over the component in the group selection area to the left and clicking the left mouse button. It is then deposited by moving the cursor into the model screen and clicking again. Components are connected by placing the cursor on the component, holding down the center mouse button and dragging the cursor to the next component, at which time the button is released. Entering of data for components is accomplished by double clicking on the desired component with the left mouse button. A dialog box will be displayed showing the type of component and the mandatory data fields to be entered, Figure 6. Generally, each component has more variables associated with it than shown in it's dialog box. The others are computed during execution of the model. Reference 2 lists the component variables associated with each component.



NAME B1SLIMIT
N 1
LL &WORLD_MODEL_CONTROL_DATA_B1SLL
UL &WORLD_MODEL_CONTROL_DATA_B1SUL

Figure 6. Dialog Box

Data assigned to each component variable can be either a numeric or point to a variable name by using the "&" operator. Variables assigned to component variables in this way are called "scope component variables". The other type of variable in Flightlab is known as a "scope free variable". The use of scope component variables adds flexibility to the model design. Variable values can be changed once in the .prolog file, vice changing it in several different components throughout the model. The use of scope component variables also negates the need of regenerating a script file each time a minor change is made. Refer to Figure 6 for an example of the "&" operator use.

Editing of components can be done by either selecting the "Edit" pull down menu or highlighting the desired component and then pressing the right button which will display a pop up menu. A component is highlighted by clicking once on it with the left mouse button. If upon selecting this pop up menu the display freezes, ensure that "Num Lock", "Caps Lock", and "Scroll Lock" options are all de-selected on the keyboard, i.e., the lights above these options are off. After turning them off the display will free up. Grouping of components is done by selecting the "Group" icon in the upper left corner of the model window, moving the cursor into the model screen, hold down the left button and as the cursor is dragged a box will be displayed. Adjust the box until it encloses all desired components and then release the button.

The model window has five pull down menus, most of which are self explanatory, but several are unique to Flightlab and will be discussed here.

Beneath the "File"pull down menu is an item called "Generate Script...". This option is chosen after a model is complete and ready to be analyzed. Upon selecting it and entering the desired file name, a text file is generated with the suffix .exc. This text file can be viewed with the editor. The script file generated cannot be executed by itself to analyze the model. It will be called by another script file, generally with a .def suffix generated by the model editor, which will generate results.

The "Group" pull down menu has three options, "Goto Parent Group", "Ungroup Selection", and "Ungroup All". These options allow the user to display the various levels which may exist in the model hierarchy, and also to move selected component(s) or entire groups of components to other levels. Model hierarchy will be discussed in detail in Chapter III.

The "Goto Parent Group", when selected, will bring up the display of the group which is one level higher in the model hierarchy. To move in the opposite direction i.e. to the display the group one level lower in the model hierarchy, double click on the desired group icon.

The "Ungroup Selection" menu item will simultaneously move the selected, highlighted, component to the next higher level and display that level. If it is desired to place a component into another existing group, drag it and place it on top of the group icon, the component will no longer be displayed when it has been successfully transferred to the new group. Double click on the group icon to display it and verify that the component has moved. These two features allow a component to be connected to another which is located in another group. For example, select a component so it is highlighted and then choose "Ungroup Selection" beneath the "Group" pull down menu. Repeat this until the component is at the desired level. Now connect the components. Return the component to the original group by placing it on top of the group icons, in succession, as described above.

The "Ungroup All" menu item will ungroup the present level displayed on the model screen and place all components in the next higher level. The group icon will remain, but it will be empty, if not needed it can be deleted.

3. Editor Window

The editor window provides a basic text editing capability. Scope files can also be executed from here if it is currently displayed in the editor window. The execute option is below the "File" pull down menu.

4. Help Window

Allows for on line help with scope commands, syntax and components.

5. Plot Window

Provides a graphical display of data as generated from the main window under scope execution.

D. MODEL ANALYSIS

Model Analysis provides an interactive menu driven format for the analysis of a model created under Model Editor and Gscope. After a short introductory message, the Main Menu will be displayed which is the top level of the analysis hierarchy. The general procedure for analyzing a model is a follows:

- Load the model
- Set initial conditions
- Prescribe desired simulation
- Execute simulation
- Plot results
- Save results, if desired

The Model Analysis program operates similarly to Gscope in that it generates scope language scripts from the choices selected in the menus. These scripts are automatically saved as text files named script.tmp , script.log , and error.log located in the directory specified by the FL_TMP environment variable. Enter the UNIX^{TM} command "env" in the shell window to display the current directory specifications. The script.tmp file is a record of the most recent analysis conducted, while script.log is a running record of all analysis conducted during the present session. The error.log file is self explanatory. When Flightlab is exited, these files are purged and will be started anew the next time Model Analysis is utilized. Either of the files can be saved under a different directory and name to be executed at a later time in the Model Analysis command line interface or in Gscope.

The choices for analyzing a model are numerous and not every scenario can be covered here. Two examples are provided to illustrate the use of the Model Analysis program. The first is a parameter sweep to get cockpit control positions for airspeeds from 0 to 150 knots in 30 knot increments. The second is a linearization, quasistatic reduction, and eigenanalysis of the model using the states [u; w; q; θ; v; p; ϕ; r].

All choices are made by entering the option number, then pressing "enter", or placing the cursor next to the option with the up and down arrow keys and then pressing "enter". In the following examples, menu titles are displayed in bold capital letters, menu items are italicized for ease of understanding. If menus have more than one page, use the "f" key to scroll to next page and "b" to scroll to previous page.

- Beginning at **MAIN MENU**
- Item 1 *Load a Model,* select local or central database
- Select desired model, when finished loading press "e" to return to **MAIN MENU**
- Item 4 *Test Operations*
- Item 2 *Analysis*
- Item 8 *Parameter Sweep*
- Item 1 *Select Parameter*
- Item 1 *Equivalent Airspeed*
- *Starting Value 0: Ending Value 150: Incremental Change 30*
- Enter "e" to return to **PARAMETER SWEEP OPTIONS**
- Item 2 *Select Analysis*
- Item 1 *Trim*
- Press "enter" twice, a "y" will be displayed to the right of *Trim*
- Enter "e" to return to **PARAMETER SWEEP OPTIONS**
- Item 3 *Start Sweep,* upon completion of calculations, enter "e" to return to **ANALYSIS SELECTION**
- Enter "e" to return to **TEST OPERATIONS**
- Item 4 *Plotting Options*

- Item 1 *Current Session*
- Item 6 *Parameter Sweep Plot*
- Item 2 *Trim Variables*
- Select trim variable, a plot vs airspeed will be displayed

To obtain a linearized eight state model from the non-linear model with the states [u; w; q; θ; v; p; ϕ; r].

- Beginning at **MAIN MENU**
- Item 1 *Load a Model,* select local or central database
- Select desired model, when finished loading press "e" to return to **MAIN MENU**
- Item 4 *Test Operations*
- Item 1 *Initialization*
- Item 1 *Configuration*
- Item 2 *Airframe*
- Item 1 *Equivalent vel in I-frame (kts)*, enter desired velocity at prompt
- Enter "e", then 1 to accept choice(s), to return to **SELECT A SUBSYSTEM**
- Enter "e" to return to **INITIALIZATION**
- Enter "e" to return to **TEST OPERATIONS**
- Item 2 *Analysis*
- Item 1 *Trim*, then select 5 to run (aircraft will at trimmed at initial conditions specified above.
- After calculations are complete, Item 3 *Linearization*
- Item 4 *Quasistatic Reduction,* then select option 2 to edit

- Item 2 *Airframe*
- Select items 22,23,5,12,23,7,10,9, upon completion enter "e", then 1 to apply and return to **SELECT A SUBSYSTEM**
- Enter "e" to return to **ANALYSIS SELECTION**
- Item 4 *Quasistatic Reduction*, then enter 4 to execute
- Item 5 *Eigenanalysis*
- Enter "e" to return to **TEST OPERATIONS**
- Item 4 *Plotting Options*
- Item 1 *Current Session*
- Item 1 *Eigenvalue Plot*
- Enter "p" to display eigenvalue plot

III. SH-60B MODEL DEVELOPMENT

Development of the SH-60B model involved using both the Model Editor and Gscope. Gscope was used to model the control system while the Model Editor was used to model the remaining systems. This approach was used because the standard model structures present in Model Editor emulated the SH-60B accurately with the exception of the control system.

Each supercomponent has associated with it four script files that are called by the .def file when loading the model. The .def file specifics will be discussed shortly. The four script files, by convention, have the suffixes .prolog, .exc, .configure, and . epilog and have as the prefix the name closely describing the supercomponent group. If selecting standard model structures in Model Editor, the four script files are already in place. If modeling a separate supercomponent, as is the case here, the script files must be created in conjunction with the model development in Gscope. The script files serve the following purposes. The execute file, control.exc, is generated in Gscope from the graphical representation of the control system as discussed in Chapter II. It is used to instance and connect components and load the data associated with each component. The prolog file, control.prolog, contains the data needed by the execute file. The epilog file, control.epilog, is where connections are made between the control system model created in Gscope an the other supercomponents in the model. Finally, the configure file, control.configure, is used to equivalence variables within the control system and also let the pilot controls, e.g. cyclic, collective, and pedals, be accessed in the Model Analysis program for aircraft dynamic response from control inputs.

Comments in any of the script files can be included and are preceded by a comment marker, //, or by using the "describe" feature in Flightlab. The describe feature allows the comments associated with data fields to be displayed when working in Gscope or Model Analysis, with a model loaded, by using the describe command. For example, the control.prolog file contains the line, k2f=6076.115/3600 "knots to feet per second conversion factor". After going to the world_model_ control_data subgroup, the command describe will display: "k2f--knots to feet per second conversion factor".

A. MODEL EDITOR PROCEDURES

Implementation of the SH-60B in the Model Editor was straight forward and followed the procedures outlined in Chapter II. Data required for the Rotor2, Airframe, FlightControl, and AeroInterference supercomponents were obtained from Sikorsky Aircraft in Reference 7. Aerodynamic data required for the Rotor1, main rotor, supercomponent was not supplied as Flightlab comes equipped with the necessary tables for the UH-60, which is identical to the SH-60B. All fuselage aerodynamic and control system data tables used in model construction are listed in Appendix B.

The flight control supercomponent was selected and saved with the model regardless of the fact that a separate flight control system was modeled. This was required to allow the .def file to execute the four script files necessary for the flight controls supercomponent. To allow the new flight control system to be called and implemented and not the one originally selected in Model Editor, the applpath.scr script file was modified. The applpath.scr file is called during execution of the .def file and is used to set a search path. In this instance the path was the location of the four script files associated with the new flight control system, control.prolog, control.exc, control.configure, and control.epilog.

Upon completion of entering the required data in Model Editor, it was saved and the script file sh60b_rigid.def was generated. A file with the .def extension is by convention a file that defines a sequence of instruction and other script files to execute [Ref. 2: p. 33]. The script file sh60b_rigid.def is included in Appendix C.

Although by examining the sh60b_rigid.def file it can be determined which super and sub components were selected and also the data and data file names associated with them, it is not an easy task. Immediately following the sh60b_rigid.def in Appendix C is an outline which contains all data entered into the Model Editor. It follows a one to one correspondence with the Model Editor tree.

The completed model can now be loaded by executing the sh60b_rigid.def in the Gscope window. The execute option is beneath the "File" pull down menu. It is recommended that the model first be loaded in Gscope vice going directly to Model Analysis and loading it. If errors are encountered during execution, they will be listed as well as the location in the .def file where execution halted. If loading a model in Model Analysis and errors are encountered, execution will stop, but no indications are given as to why.

B. CONTROL SYSTEM MODELING WITH GSCOPE

1. Model Hierarchy

A Flightlab model is built using a hierarchial technique. Simple sub-systems are built, tested on their own and then combined to form the complete model [Ref. 3: p. I-1]. This method allows for a modular design where sub-systems can be easily replaced with variations of that sub-system, e.g., a main rotor system with four blades vice three.

At the lowest level in the hierarchy is a group. A group is made up of a series of components and given a name which describes the sub-system of the aircraft. Groups can exist within groups. Obviously, a group which contains another group or groups is one level higher in the hierarchy. Many sub-system groups are created and linked to form the complete aircraft and collectively called the "model" group. The model group is, in turn, contained within the "world" group, which is the top level of the hierarchy.

The remainder of this chapter describes the construction of the SH-60B control system and how it is integrated into the part of the model created under Model Editor. The presentation will follow a top down order beginning at the top level, which is the world group. This is the way models are presented when first opened in Gscope. To aide in ease of reading and understanding, all component names will be *italicized*, group names will be **bold**, names assigned to components will be CAPITALIZED and script file names will be <u>underlined</u>. Flightlab does not have an option to name the components as shown in the following figures. The names were inserted using a separate graphics utility.

2. World Group

The top level in any model is the **world** group and for the SH-60B model contains only the **model** group. If modeling an entire aircraft, the **world** group would also contain the *atmosystem* and *aerosystem* components [Ref. 2: p. 11]. When selected, the **model** group has one group beneath it, the **control** group. Beneath it are the **Sensors, Stabilator,** and **FlightControls** groups. Figure 7 shows the progression of the SH-60B model hierarchy.

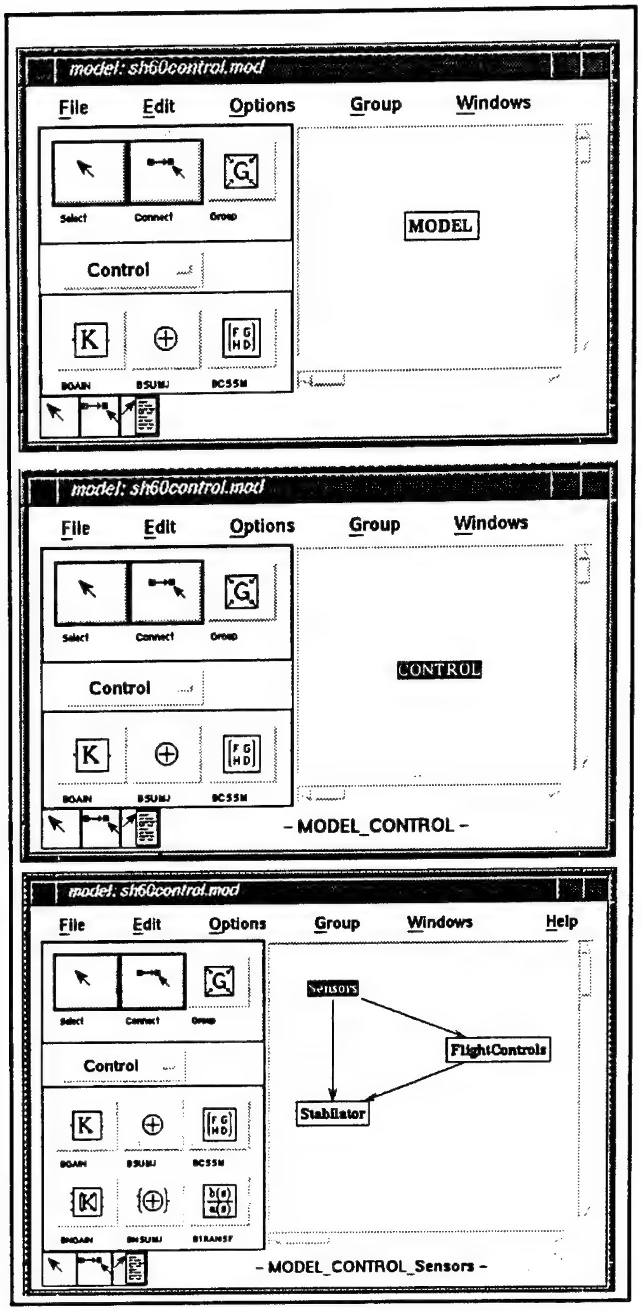


Figure 7. SH-60B Model Hierarchy

3. Sensors Group

The **sensors** group is used to provide state variable input to the various control groups during execution of the model. The **Sensors** group is shown in Figure 8. At the top of Figure 8 are two *source-transfer function-sink* combinations and a single *source* component. The *sink* components labeled ALPHA and BETA output the models angle of attack and sideslip, respectively. The *transfer function* components are first order low pass filters which introduce a very short time delay into the calculations to prevent "cyclic

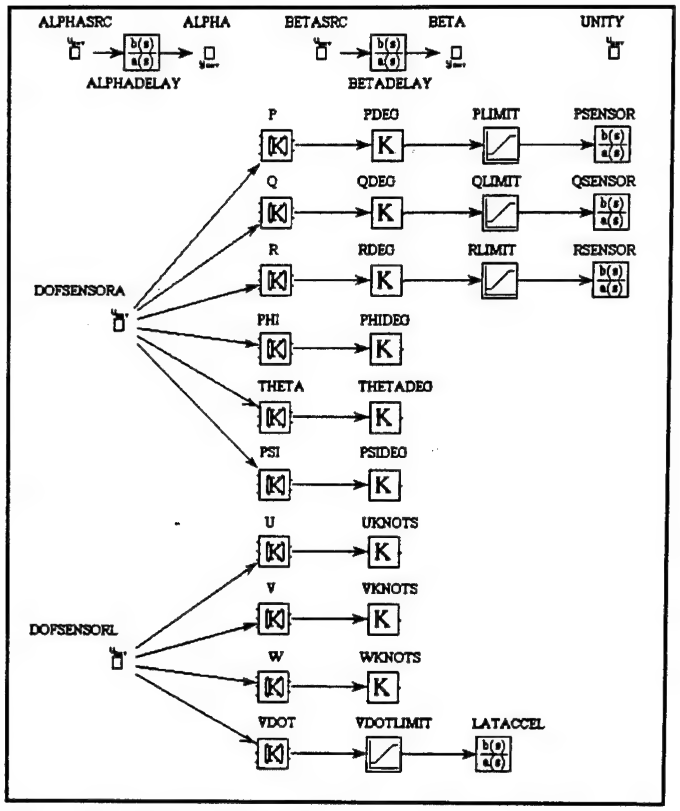


Figure 8. Sensors Group

data dependency" errors. The *source* component UNITY provides a constant source of one for input into various parts of the control system.

The aircraft euler angles and euler angle rates are input into the **Sensors** group via the DOFSENSORA *source* component, while the aircraft body velocities and accelerations are input via the DOFSENSORL *source* component. Since construction of the control system is in Gscope while the remainder of the model was completed under Model Editor, the connections of DOFSENSORA and DOFSENSORL to the **airframe** group must be accomplished in the control.epilog file.

The *source* components each are connected to *bngain* components that act as demultiplexers to select one of the degrees of freedom from the *source* component. Each *blimit* component is connected to a *bgain* component to convert the quantity to the desired units, degrees, degrees per second, or knots. Next, the *blimit* components PLIMIT, QLIMIT, RLIMIT, and VDOTLIMIT set upper and lower limits of angular rates and lateral acceleration for input into their respective transfer functions. The *transfer function* components labeled PSENSOR, QSENSOR, RSENSOR, and LATACCEL are all second order filters which represent the sensor dynamics plus filters in the automatic flight control system, (AFCS), and SAS computers.

4. Stabilator Group

The incidence angle of the stabilator, horizontal tail, of the SH-60B is programmable, the control system for which is presented in Figure 9. The incidence angle is set by the control system so as to maintain a nearly level pitch attitude as a function of forward airspeed pitch rate, lateral acceleration and collective position, with forward airspeed having the

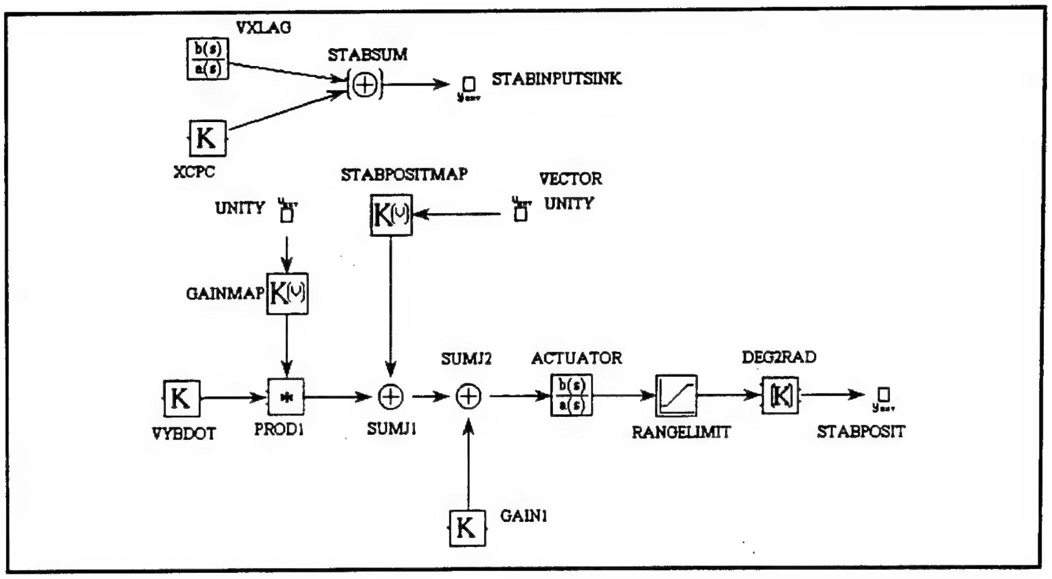


Figure 9. Stabilator Group

greatest influence on setting the incidence angle. At low airspeeds, below approximately 30 knots, the stabilator incidence angle is set at 40 degrees, leading edge up, and decreases with increasing airspeed.

Forward airspeed in knots is input into the *btransf* component VXLAG in Figure 9. Collective position in inches is input into the *bgain* component XCPC. The XCPC gain multiplies collective position in inches by ten to convert the quantity to percent collective position. These two quantities are then combined to form a column vector by the *bsumj* component STABSUM. The column vector is in turn input into the *bsink* component STABINPUTSINK.

The *bvgain* component STABPOSITMAP is a two dimensional lookup table with forward airspeed and percent collective position as the two parameters, which are obtained from STABINPUTSINK. The output of STABINPUTSINK is not directly connected to STABPOSITMAP, but is made in the control.configure file by equivalencing the input and output variables of each component. The command: world_ model_control_stabilator_ stabpositmap_var=&world_model_control_ stabilator_stabinput_y, sets the output of STABINPUT equal to the input of STABPOSITMAP. The *bvgain* component is generally used as a scalar or matrix gain block in the feedback loop of a control system. In this case the input would be made by directly connecting the components in the model window. When the *bvgain* component is used as a two dimensional lookup table, its input is obtained via its "var" parameter. The VECTOR UNITY block supplies a constant input vector of [1;1]. The second *bvgain* component GAINMAP operates in the same fashion except it is a one dimensional lookup table with the output of VXLAG as its input parameter.

Lateral acceleration is input into system via the *bgain* component VYBDOT. The gain has a value of one and was used to help clarify this input. Pitch rate is input via the *bgain* component GAIN1. The *btransf* component ACTUATOR is second order and represents the dynamics of the actuator which drives the stabilator. The *blimit* component RANGELIMIT sets the upper and lower limits of travel at 40 and -8 degrees respectively. The value is then converted to radians and sent to the *bsink* component STABPOSIT.

The output of STABPOSIT is equivalenced to the *chinge* component EITAIL in **hstab1** and **hstab2** groups in the control.epilog file. Two connections must made because Flightlab models the stabilator as two independent groups which can operate independent of each other. The left and right sides of the stabilator are mechanically connected in the SH-60B, and therefore operate as one.

5. Flight Controls Group

The **FlightControls** group consists of the **Longitudinal, Lateral, Collective,** and **Directional** sub-groups, Figure 10, which model the four control axes. These sub-groups all operate similarly so only the **Longitudinal** sub-group will be explained, which is shown in Figure 11.

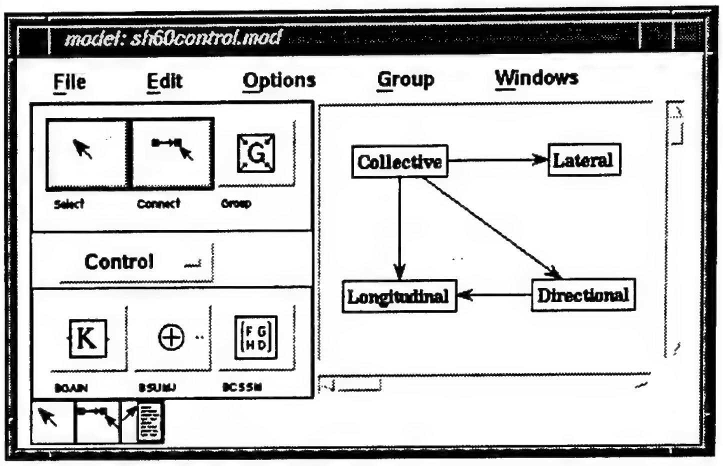


Figure 10. Flight Controls Group

a. Longitudinal Group

Excluding the four sub-groups **PitchAnalogSAS, PitchDigitalSAS, PitchSASInput,** and **PBA**, this sub-group models the mechanical portion of the longitudinal axis. The longitudinal control system starts with two *source* components XBTRM and XB, which represent the longitudinal cyclic stick position in inches. The XBTRM component is set to a value determined by the trim control matrix which is calculated during the trim routine for the model [Ref. 2: p. 27]. The XB component is used when flying or performing an analysis on the model, otherwise it is set equal to zero. The two values are then summed

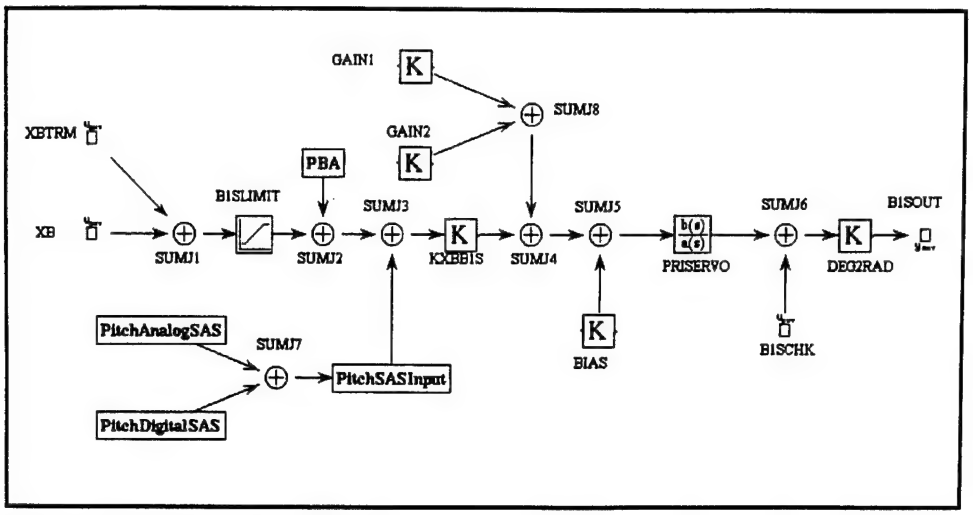


Figure 11. Longitudinal Group

and sent through the *blimiter* component B1SLIMIT which sets the upper and lower limits of longitudinal cyclic travel at 10 and 0 inches respectively. Inputs from the stability augmentation system, SAS, and pitch bias actuator, PBA, are then added. The *bgain* component KXBB1S converts inches of longitudinal cyclic position to degrees of main rotor longitudinal pitch angle, commonly called B1S.

The collective and directional control axes supply inputs to the longitudinal control axis via the *bgain* components GAIN1 and GAIN2 respectively. This is known as control mixing. The lateral and directional axes also have control mixing, although only from the collective axis. The purpose of control mixing is to decouple the helicopter's response by feed forward of control inputs. For example, tail rotor pedal motion is input into the longitudinal axis to counter act the pitching moment from the canted tail rotor. Control mixing is implimented directly in the control system via the mechanical linkages.

The sum of collective and tail rotor control mixing inputs is summed and in turn summed at SUMJ4 and again summed at SUMJ5 with the control axis bias. The bias for each control axis is the angular position of the control surface with the cockpit cyclic, pedals and collective in their reference control positions. The SH-60B reference control positions are; full aft longitudinal cyclic, full left lateral cyclic, full down collective, and full left pedal. For example, the longitudinal axis has a bias of 9.7754 degrees. When all cockpit controls are placed in their reference position, the longitudinal cyclic pitch is 9.7754 degrees. The output is then passed through the *btransf* component PRISERVO, which represent the dynamics of the hydraulically actuated primary servos in the helicopter.

The next portion of the control system sums the output of the PRISERVO with the *source* component B1SCHK. This component is used during development of the control system and for inputs directly into the longitudinal pitch angle of the main rotor. The control position is converted to radians by the DEG2RAD component and finally input to the *sink* B1SOUT. The output of B1SOUT is equivalenced to "b1s" in the control.configure file. This allows the commanded longitudinal cyclic pitch angle to be input into the **rotor1** group.

b. Stability Augmentation System

The lateral, longitudinal and directional control axes each have digital and analog SAS channels. The collective SAS channel contains the barometric altitude hold, radar altimeter hold, and coupler which are autopilot functions, and are not relevant to flight dynamic modeling [Ref. 8: Appendix A p. 5]. The analog and digital SAS implemented here are not the complete systems present in the SH-60B, but only approximations. The full model is proprietary property of Sikorsky Aircraft and therefore could not be supplied for this thesis work.

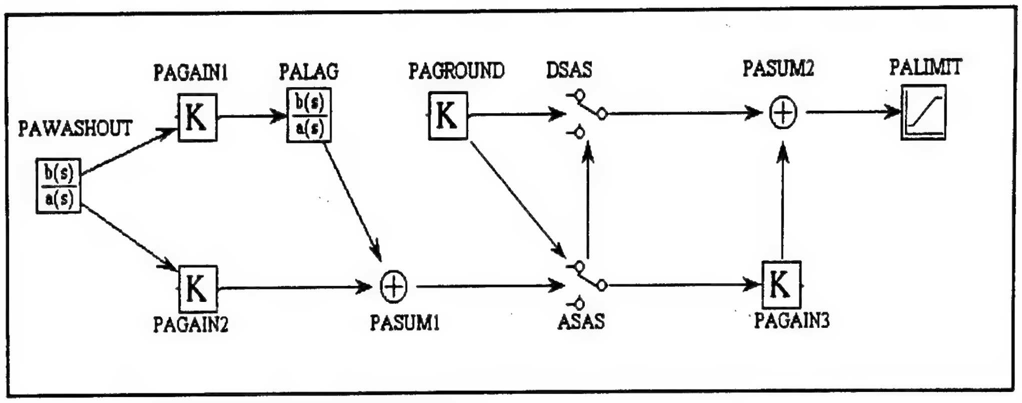


Figure 12. Pitch Analog SAS Group

The pitch analog SAS group in Figure 12 shows the basic pitch SAS architecture. Pitch rate is input to the channel via the *btransf* component PAWASHOUT.

The pitch rate signal is then sent through a proportional gain path, PAGAIN2, and a lagged path, PALAG. The proportional path provides damping, while the lagged path adds stability [Ref 8: Appendix A p. 5]. The five components PAGROUND, ASAS, DSAS, PASUM2, and PAGAIN2 make up a logic circuit that determines the gain applied to the signal depending on whether the analog and digital SAS channels are on or off. If the digital SAS channel is off, the gain applied to the analog SAS signal is 2, and vice versa. This allows an increase in control response to help offset the one system being disengaged. Although the gain is doubled, the range limits remain the same. If both systems are on the gain is 1, and if both are off the gain is 0. The PALIMIT component limits the maximum authority of the pitch analog SAS to +/- 6.0 percent, the output of which is summed with the output of the pitch digital SAS, refer to Figure 11.

The pitch digital SAS channel is very close in construction to the analog, so reviewing it here would be of little value. Instead, the yaw digital SAS channel will be presented. Figure 13 shows the basic architecture of the yaw SAS channel. Yaw rate is fed into the yaw SAS channel via the YDWASHOUT component and then sent through a proportional and lagged path, for the same reasons as the pitch analog SAS channel.

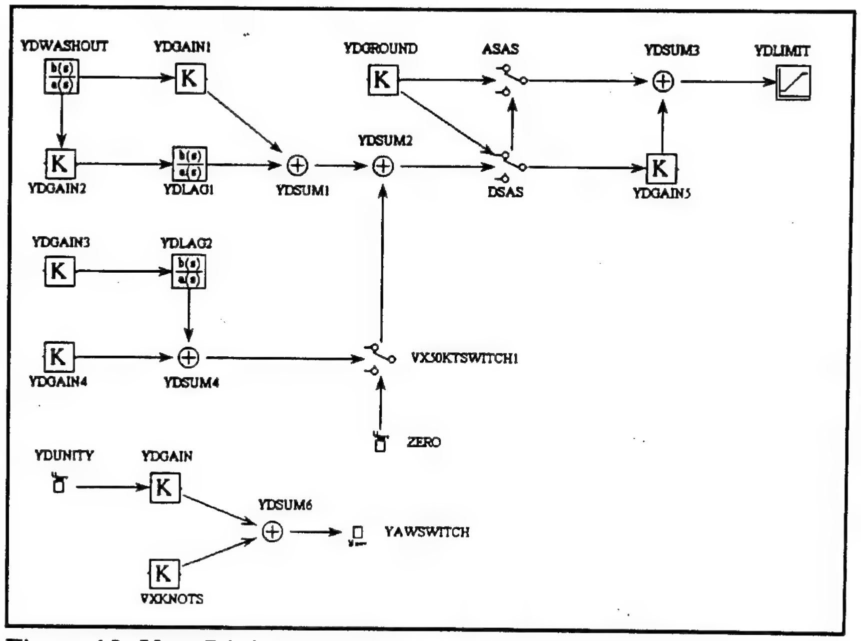


Figure 13. Yaw Digital SAS Group

The major difference between the yaw SAS channels, both the digital and analog, and the other channels, is a velocity dependent switch which causes a change in control laws at 50 knots. Below 50 knots, the yaw rate is fed to proportional and lagged channels to enhance yaw damping and stability. Above 50 knots, the yaw channel is used to provide turn coordination. In this case, yaw rate, roll rate, and lateral accleration are used as feedback to improve the flying qualities.

Lateral acceleration is fed in through the *bgain* component YDGAIN3 then to a lag filter, YDLAG2. Roll rate is input via the YDGAIN4 component and then summed with the lateral acceleration signal at YDSUM4. The *bdisw* component VX50KTSWITCH1 serves as the velocity dependent switch. When body velocity in the x-direcition, i.e., forward velocity, is greater than 50 knots the switch is set so as to allow the signal to pass. If less than 50 knots the switch is set to pass a value of zero, which is input via the *bsource* component ZERO. The group of components at the bottom of Figure 13 serve as a logic circuit to set the velocity switch. A value of one is multiplied by a gain of 49 in the YDGAIN component and then subtracted from the x-direction body velocity, input via the VXKNOTS component, at YDSUM6. The quantity is then input to the YAWSWITCH component. If this quantity is less than 1, i.e., velocity less than 50 knots, the VX50KTSWITCH1 is set to pass a signal of zero, if greater than 1, it passes the summed lateral acceleration and roll rate signal. The connection between the YAWSWITCH and VX50KTSWITCH1 components is accomplished in the control.configure file. The group of components in the upper right of Figure 13 serve as a logic circuit to set the applied gain to the yaw SAS signal, just as in the pitch analog SAS discussed above.

The actual turn coordination logic and control law in the yaw axis is considerably more complex than modeled here. The full system is considered proprietary by Sikorsky Aircraft, and the system modeled here was provided as a functional equivalent to the actual system.

c. Pitch Bias Actuator Group

The pitch bias actuator, PBA, shown in Figure 14 is a second pitch stability channel in addition to the analog and digital SAS channels. The PBA is installed in the SH-60B to provide a positive stick position gradient with airspeed. In the bucket airspeed region, where longitudinal stick gradient is shallow or slightly reveresed, the PBA applies aft longitudinal cyclic requiring forward stick to compensate and creating a positive stick position gradient. In addition, pitch rate and pitch attitude are used as feedback to enhance the longitudinal damping and stability respectively.

Pitch rate and theta are input via the *blimiter* components PBAQLIMIT and PBATHETLIMIT respectively. Each is multiplied by a gain and then summed at SUMJ1. Forward velocity is input via the *bvgain* component PBA3MP which is a one dimensional lookup table that multiplies the velocity by a gain. This output is then summed with the pitch rate and theta quantities and again summed with a bias of -4.9 percent.

The portion of the diagram between SUMJ4 and PBAGAIN3 represents the dynamics of the actuator. The first limiter component limits the rate of the actuator to $^{+}/_{-}30$ percent per second. The second limiter limits actuator authority to $^{+}/_{-}15$ percent. The PBASWITCH component allows the PBA to be disconnected from the model in order to allow an open loop analysis. The PBA cannot be secured in the SH-60B as shown here. The last two components combine to convert percent of longitudinal cyclic stick position to inches. Refer to Figure 11 for the location of the pitch bias actuator input into the longitudinal control axis.

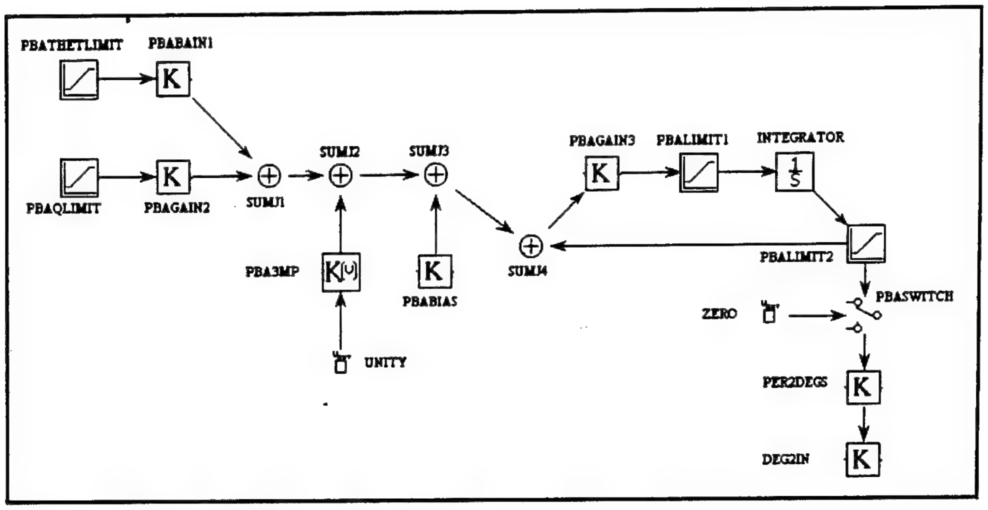


Figure 14. Pitch Bias Actuator Group

IV. SH-60B MODEL ANALYSIS

A. COMPARISON WITH GENHEL RESULTS

All of the analyses conducted on the SH-60B for comparison with GenHel utilized two script files, trimsweep.scr and powerreq.scr and were done at a gross weight of 19462 pounds. The trimsweep.scr file outputs cockpit control position, pitch and roll attitude, and stabilator incidence angle versus airspeed. The powerreq.scr file gives main rotor and total power required for a user specified weight and airspeed range. The outputs of trimsweep.scr can all be obtained via the parameter sweep menu in Model Analysis with the exception of stabilator incidence angle. Main rotor and total power required versus airspeed cannot be obtained in Model Analysis. Both script files are included in Appendix D.

The first comparison between Flightlab and GenHel results was a level flight trim sweep of cockpit control positions versus airspeed, Figure 15. The y-axis limits on each plot are full travel of each control axis.

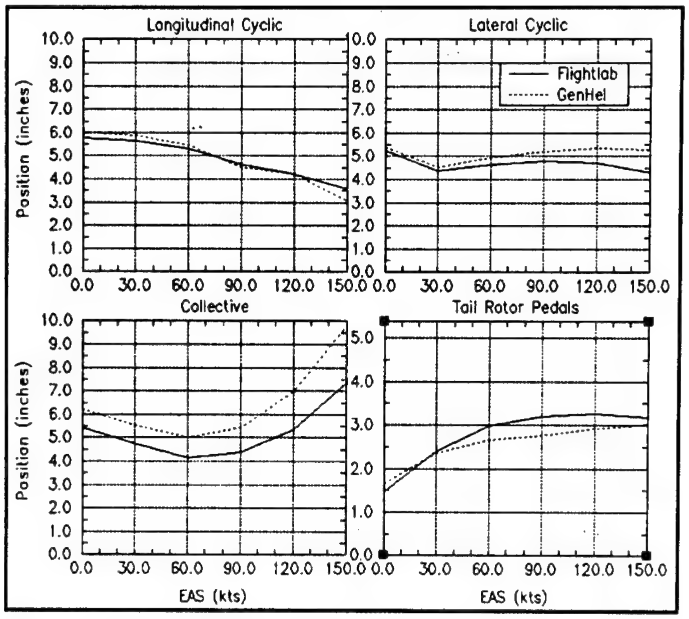


Figure 15. Level Flight Trim Sweep

Longitudinal, lateral and pedal control positions all show good comparison, but the collective differs by a nearly constant bias of one inch. This is ten percent, relative to full travel, and was initially deemed unacceptable until a comparison of main rotor power versus airspeed was conducted, Figure 16.

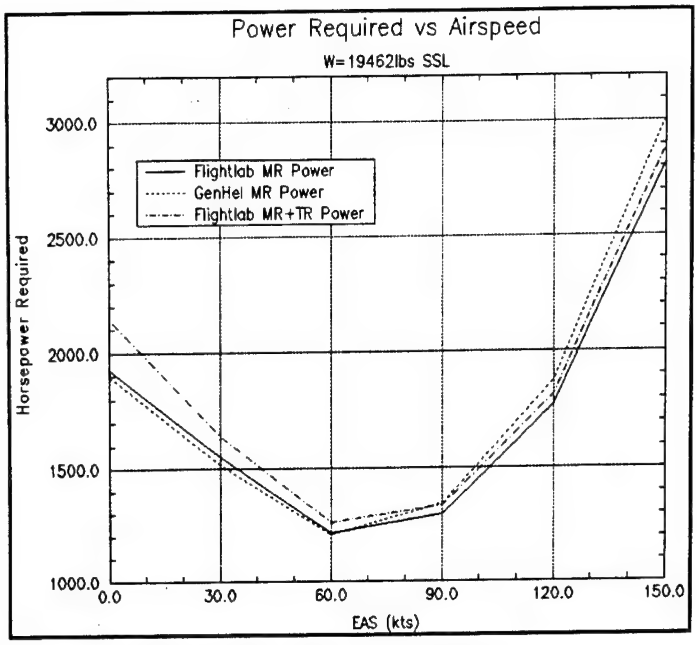


Figure 16. Power Required vs Airspeed

The comparison between Flightlab and GenHel main rotor power required was very good. From 0 to 80 knots the difference is negligible, above 80 knots the difference increases slightly to 6 percent at 150 knots. Referring back to Figure 15, collective position versus airspeed, if the Flightlab curve was moved up so as to be coincident with the GenHel curve at 60 knots, the trends between collective position and power required would be identical. At low airspeeds Flightlab is slightly higher, both converging to the same value at 60 knots and then the difference increasing linearly up to 150 knots.

Although the collective position differed, the more significant parameter of main rotor power agreed well. Flightlab tail rotor plus main rotor power, i.e. total power, is also included Figure 16. It shows the expected trend of tail rotor power required decreasing with increased airspeed as the vertical tail becomes more effective at providing anti-torque to the main rotor.

The third parameter to compare Flightlab and GenHel models was stabilator incidence versus airspeed, Figure 17. Up to 30 knots the results are identical due to the

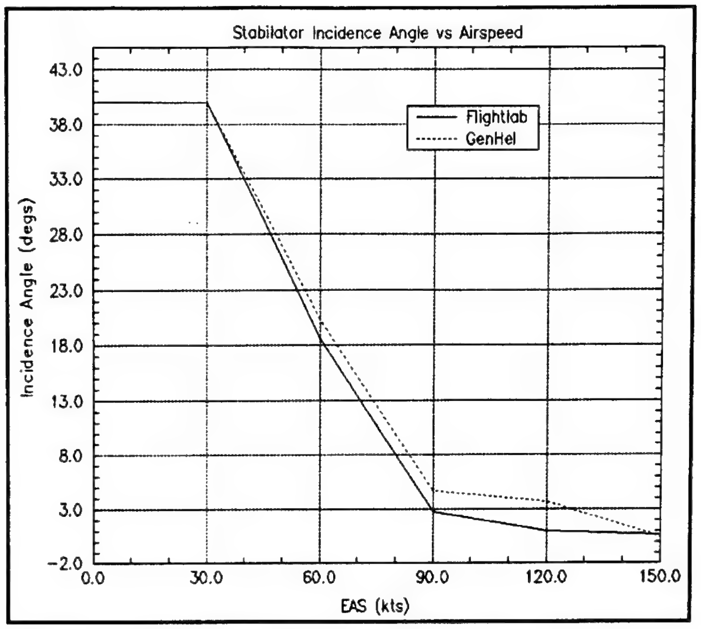


Figure 17. Stabilator Incidence Angle vs Airspeed

that the maximum travel limit of the stabilator is set to 40 degrees. The difference between the two increases and then converges to the same value at 150 knots. Part of the difference is attributed to the fact that one of the inputs to the stabilator control system is collective position. As discussed above, the collective position versus airspeed differed by a nearly constant bias of one inch. A second run of stabilator position versus airspeed was conducted with a bias of one inch added to the collective input into the stabilator control system. The results are shown in Table V, along with original values obtained and the GenHel Results.

The results with the bias are very close to the GenHel results. The difference between collective position for the two programs cannot be explained as GenHel was not available for use. It is clear though that the difference between the two are due to computational differences within each program as the output variables, power required and stabilator incidence angle differed by only a small amount when taking into account the collective bias.

EAS (kts)	GenHel	Flightlab w/o bias	Flightlab w/ bias
0	40.0	40.0	40.0
30	40.0	40.0	40.0
60	20.3	18.58	20.13
90	4.7	2.76	4.33
120	3.7	0.98	2.58
150	0.5	0.64	0.64

Table V. Stabilator Incidence Angle In Degrees vs Airspeed

The fourth comparison between Flightlab and GenHel was pitch and roll attitude versus airspeed, Figure 18.

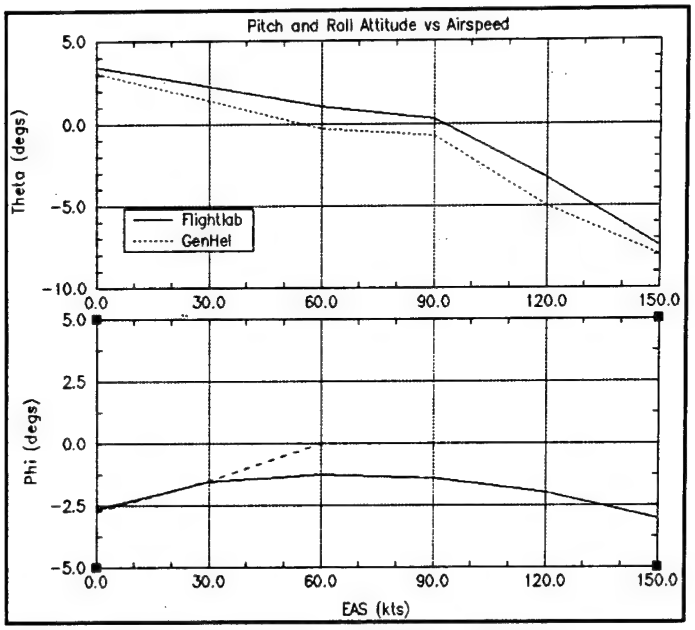


Figure 18. Pitch and Roll Attitude vs Airspeed

Flightlab results showed a slightly higher pitch attitude, theta, over the airspeed range than GenHel. Again, part of the difference can be attributed to the collective bias. The stabilator incidence angle, which is a factor in pitch attitude, was improved with the collective bias accounted for, and so was pitch attitude, but to a smaller extent. Table VI summarizes the results. Roll attitude, phi, compared well up to 30 knots, but diverged from the GenHel results. From 60 to 150 knots GenHel showed a level attitude, where Flightlab had a left roll angle.

EAS (kts)	GenHel	Flightlab w/o bias	Flightlab w/ bias
0	3.10	3.48	3.48
30	1.42	2.28	2.28
60	-0.30	1.07	0.79
90	-0.71	0.30	-.11
120	-4.98	-3.27	-3.84
150	-7.98	-7.45	-7.46

Table VI. Pitch Attitude In Degrees vs Airspeed

B. CLOSED LOOP ANALYSIS

The control systems of the model are comprised of the analog and digital SAS channels and the pitch bias actuator. State variable feedback is used in these systems to provide a stable model, which otherwise is unstable. Both characteristics will be shown in this section and the next. All of the analyses therefore conducted in this section are with the three systems on, making the model closed loop. All analyses were conducted at a gross weight of 19462 pounds, 90 knots equivalent airspeed and at standard sea level conditions. The coordinate system for the model is defined with the positive x-axis out the nose of the aircraft, positive y-axis out the right side and positive z-axis down. Angular displacements are defined in accordance with the right hand rule.

In Flightlab the nonlinear model can be linearized and expressed in the standard form

$$\dot{x}=\tilde{A}x + Bu \qquad (1)$$

$$y = Cx + Du \qquad (2)$$

The $\tilde{A}$ notation is used to denote the closed loop plant. Since state variable feedback is implemented in the model;

$$\tilde{A}=A-BK \qquad (3)$$

where A and B are the open loop plant matrices and K is the gain Matrix. The stability of the model can then be determined by finding the eigenvalues of the $\tilde{A}$ matrix.

The Model Analysis program within Flightlab was used to linearize the model. Prior to executing the linearization though, the desired conditions must be set and the model trimmed. The result yielded a $\tilde{A}$ matrix that was 77x77, i.e., 77 states, and a B matrix that was 77x4. The columns of the B matrix correspond to the four cockpit inputs, longitudinal and lateral cyclic, collective and tail rotor pedals. The eigenvalues of the model were all in the left hand plane indicating that it was stable. To confirm this, standard control input tests were conducted for each of the control axes and the time history response of the aircraft states recorded. The nonlinear and linear response are presented for each test. The linear responses shown for each of the tests are that of the full 77 state model.

The first test was a longitudinal cyclic impulse of one inch lasting for three seconds with a one second rise and fall time. The time response is given in Figure 19. The linear

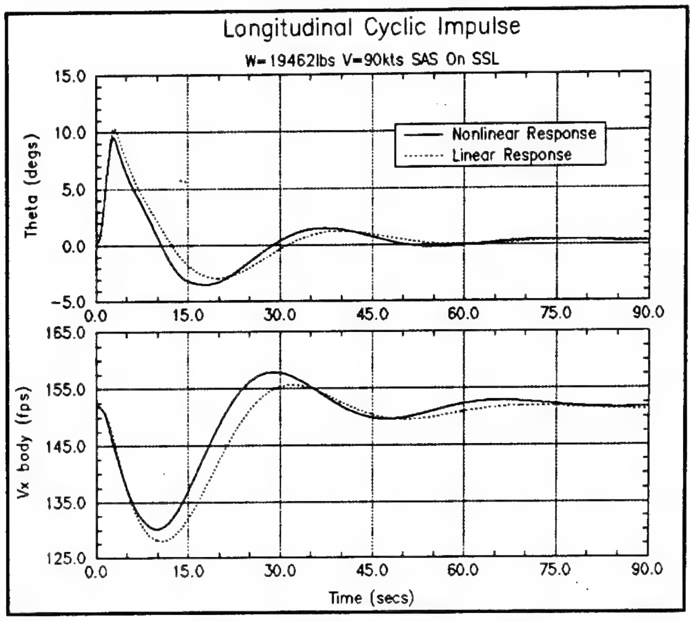


Figure 19. Longitudinal Cyclic Impulse Response

and nonlinear response correspond very well and dampen out to a steady state value in both pitch and forward velocity by approximately 90 seconds.

The second test of aircraft stability was a lateral impulse of the same description as for the longitudinal test. Refer to Figure 20 for the time history response. The linear response has a smaller amplitude in roll and roll rate and dampens out quicker than the nonlinear response. The response of the aircraft pitch angle, theta, is also included in the figure to demonstrate the roll to pitch coupling of the SH-60B. As a right roll is introduced the aircraft nose pitches up and when the lateral cyclic is returned to the trimmed position the nose pitches down. The coupling is so strong that the initial response in pitch is nearly as high as the roll angle.

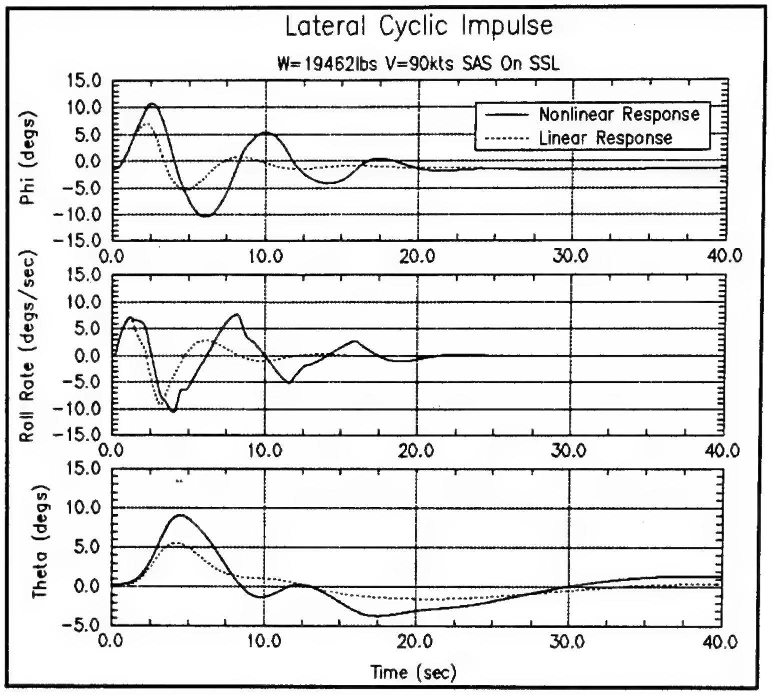


Figure 20. Lateral Cyclic Impulse Response

To test the directional stability of the model, a tail rotor pedal doublet of one inch lasting for three seconds with rise and fall time of one second and a zero delay between each impulse was used. Figure 21 shows the time response of the aircraft to such an input. Again the linear and nonlinear responses are very close with the linear response having a smaller amplitude and damping out quicker than the nonlinear response.

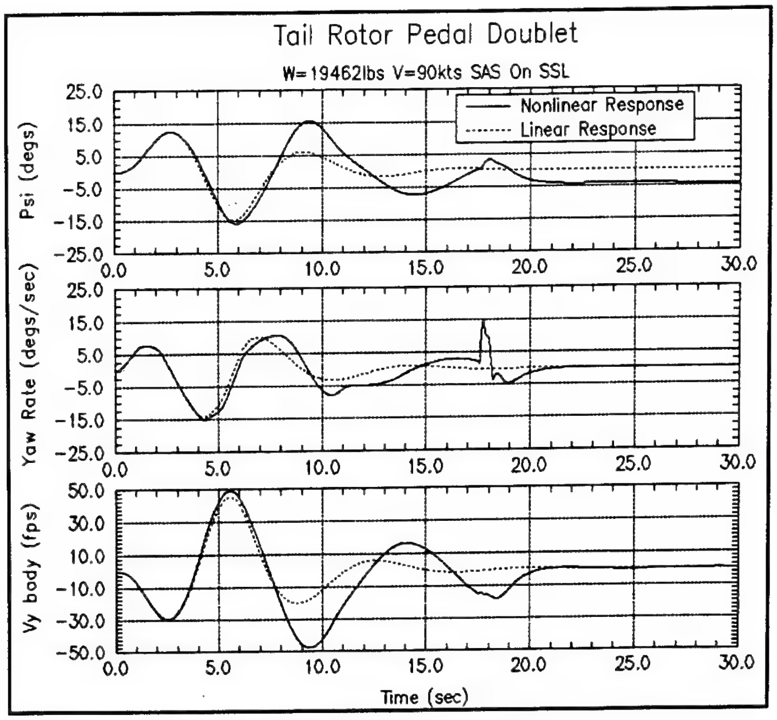


Figure 21. Pedal Doublet Response

The final test of the closed loop response was a collective doublet with the same dimensions as the pedal doublet. The response of yaw angle, yaw rate and vertical body velocity are shown in Figure 22. In addition to demonstrating the stability in this axis it shows the reaction of the aircraft to an increase and then decrease in collective setting. The coordinate system for the model has the negative z direction defined as up, therefore the plot of Vz body shows the aircraft initially climbing. The corresponding increase in torque produced by the main rotor with an increased collective setting causes a yaw to the right, defined as positive. Figure 22 actually shows the yaw and yaw rate initially going in the opposite direction as would be expected. This is not a discrepancy in the results, but a trait of a non-minimum phase system. If at least one pole or zero of the transfer function is in the right half plan the system is called non-minimum phase. As none of the poles were in the right half plane the transfer function of collective to yaw and yaw rate must have at least one zero in the right hand plane. Importing the matrices into MATLAB^{TM} and converting them to transfer functions of collective to yaw and yaw rate, it was discovered that each transfer function had one zero each in the right half plane, thus explaining the yaw and yaw rate response.

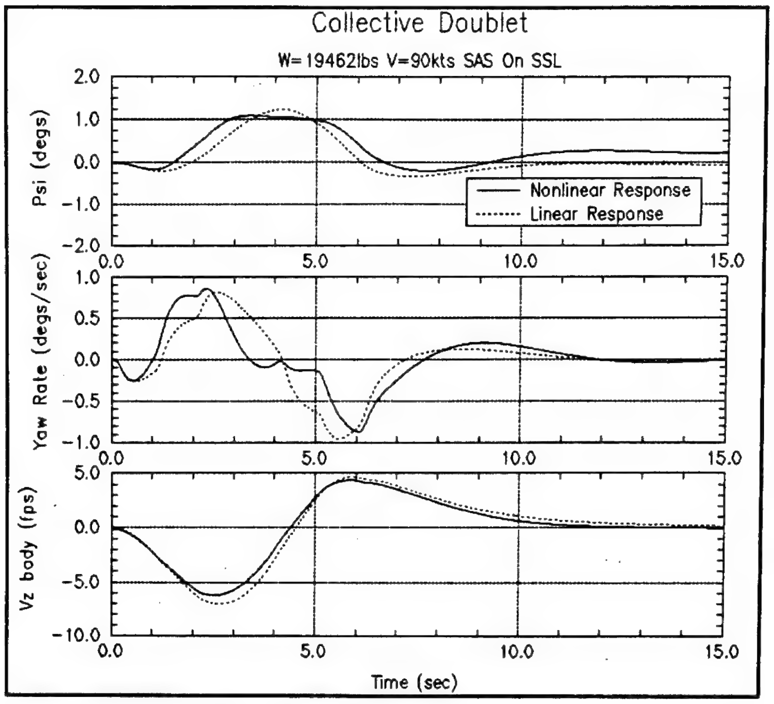


Figure 22. Collective Doublet Response

The transfer function of collective to Vz body also had a zero in the right half plane, but its effects on the response of Vz body is negligible. This is because the zero was at 118.3 where for yaw rate it was at $4.65\text{x}10^6$. The same can also be said for psi, yaw angle, which had a zero at $1.60\text{x}10^4$.

A linear frequency response analysis was also conducted on the closed loop system in the longitudinal and lateral axes. Flightlab can also do a nonlinear frequency response analysis, but the differences between the linear and nonlinear time history plots were so small, it was assessed that the linear frequency response would also give accurate results.

The frequency response analysis can be conducted from the Model Analysis program, but it does not give the user the option to select the frequency range of interest. The frequency range is hard coded in with a range from 1.6 to 100 rad/sec. A second drawback is that the preset frequency range is not logarithmically spaced. In order to remedy these two drawbacks, the file udf.fun was copied into the users home directory and modified. The file is located in the directory $FL_DIR/flightscope/analysis. The command: ww=logspace(w1,w2,num), was entered after the original assignment command. The syntax for the logspace command: w1 is the lowest and w2 is the highest frequency in the range and num is the number of elements. The function "Linear FrequencyRespons"" is included in Appendix D. Before executing the frequency response analysis in Model Analysis, but after loading the model, the command: exec("udf.fun",1), must be executed in the command line interface. This allows the udf.fun file in the users home directory to be used in the analysis, vice the original one.

Figure 23 is a frequency response plot of Vx body and pitch angle, theta, for a longitudinal cyclic input. At the low frequency spectrum the magnitude of theta is below

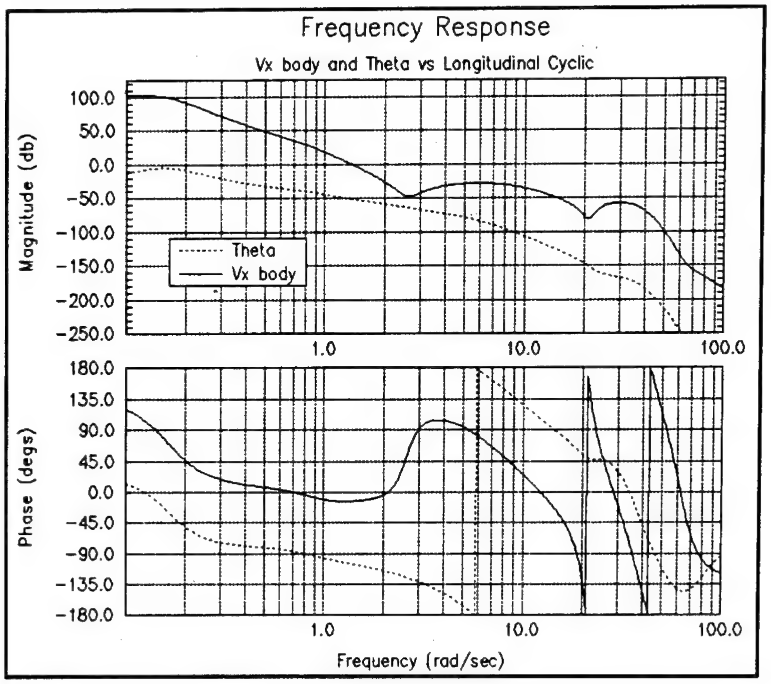


Figure 23. Frequency Response From Longitudinal Cyclic Input

0 db, while that of Vx body starts at 100 db and shows a large amount of damping as evidenced by the 80 db/decade decrease in gain. The magnitude crosses the 0 db line at approximately 1.5 rad/sec. The interpretation of this plot, as it applies to the physical

description of the helicopter motion, is that large changes in forward velocity with little change in pitch angle are experienced below inputs of 1.5 rad/sec. Above 1.5 rad/sec both forward velocity and pitch angle response are highly damped. This type of response is not observed in fixed wing aircraft, i.e. the magnitude of the pitch response is higher at lower frequencies. Part of the reason is due to the fact that longitudinal, and lateral, inputs are made into the main rotor, vice directly into the aircraft body as in the case for a fixed wing airplane. It is obvious that an input into a rigid body will have a greater response magnitude vice an input into a dynamic system, i.e., main rotor, coupled to the rigid body. The magnitude of the airspeed response remains high at low frequencies because the main rotor forward and aft pitch, which does have a high gain at low frequencies, acts as a "brake" to affect airspeed with little damping.

Figure 24 is a frequency response plot for Vy body and roll angle, phi, versus lateral cyclic. The exact same trend is noted as in the longitudinal case which should be expected. Figure 25 is a frequency response plot of pitch angle versus a lateral cyclic input. Notice that the magnitude of theta is very close to the magnitude of roll angle for a lateral cyclic input due to the cross coupling of the two. This aspect was also discussed in the time history response to a lateral input.

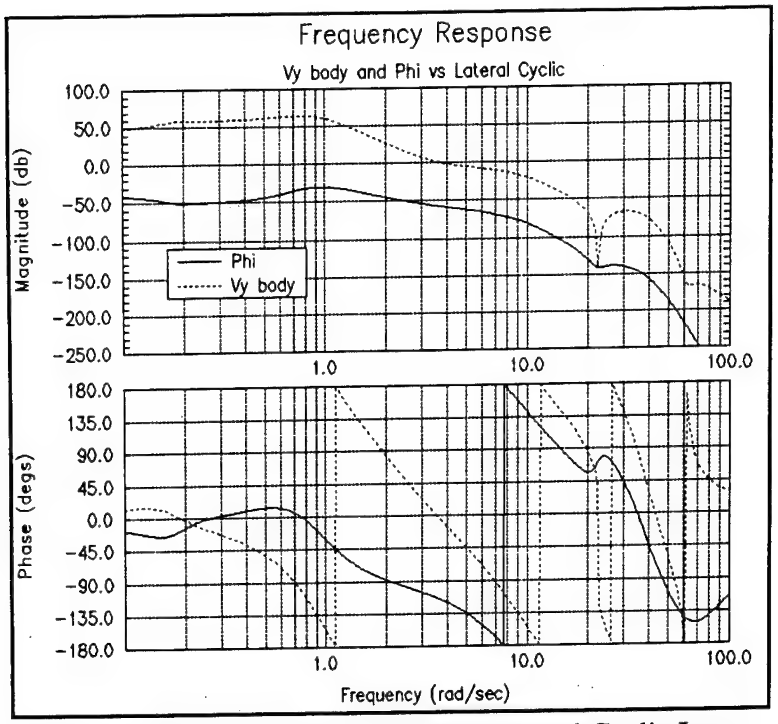


Figure 24. Frequency Response From Lateral Cyclic Input

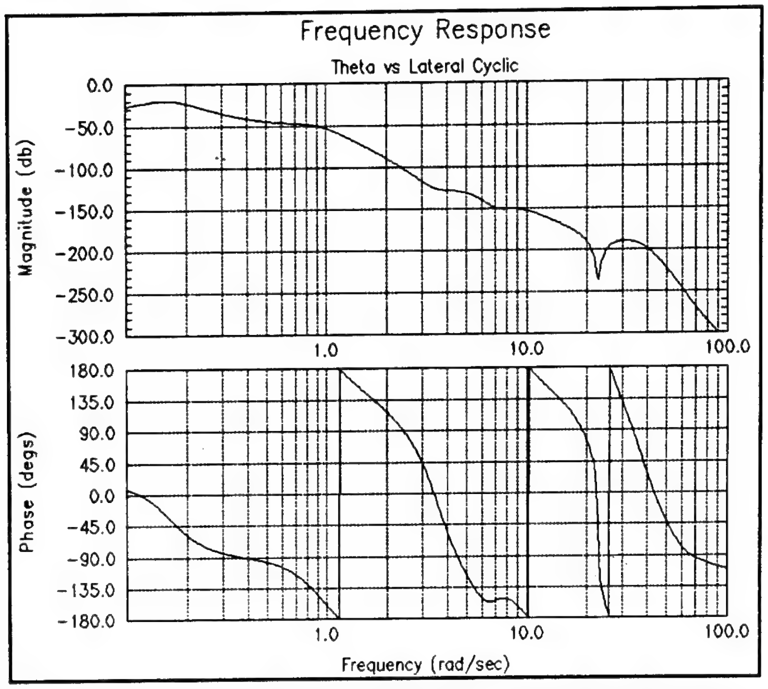


Figure 25. Frequency Response From Lateral Cyclic Input

C. OPEN LOOP ANALYSIS

The model is open loop with the analog and digital SAS and pitch bias actuator disengaged. All three systems can be turned off by setting their flags from one to zero in the "initialization/configure/flight controls" menu within Model Analysis. After loading it and setting the initial conditions to 19462 pounds gross weight and an equivalent airspeed of 90 knots the model was linearized. The result was a 77 state plant, the same as with the closed loop, but it had three unstable eigenvalues. This 77 state linear model was then reduced to a model with six degrees of freedom and eight state variables, [u; w; q; θ; v; p; ϕ; r]. The reduced order plant also had three unstable eigenvalues, almost exactly in the same location as the full 77 state linear model. Figure 26 is an eigenvalue plot of the open loop reduced order model. The A and B matrices and the eigenvector and eigenvalues for the reduced order model are given in Appendix E for airspeeds of 0, 30, 60, 90, and 120 knots.

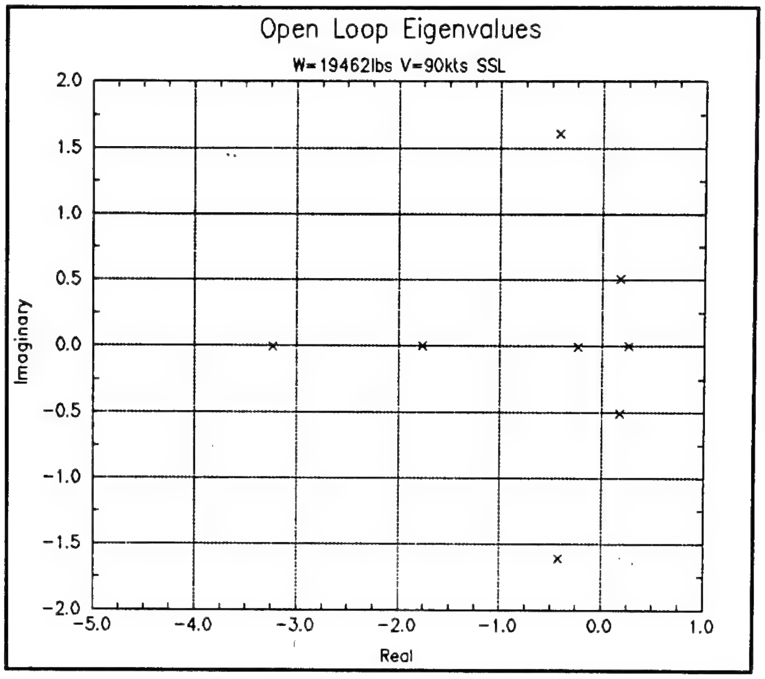


Figure 26. Open Loop Eigenvalues

Figure 26 shows a pair of complex unstable roots and one real unstable root. It will be shown that the real root is associated with the longitudinal axis, and that the complex roots with the lateral axis.

In the comparison between the nonlinear and linear response in the previous section, the full 77 state plant was used. This section will use the reduced order plant for the control input tests for two reasons. It has already been shown that the full linear model gives good results when compared to the nonlinear response, so it will be now assessed how the reduced order compares with the nonlinear. Also, to determine which modes are associated with the unstable eigenvalues in the eight state model.

The first test was a longitudinal impulse of one inch for a duration of .025 seconds. A short duration input was used to simply perturb the model about the trimmed condition and observe the response. The closed loop analysis required inputs of a longer duration due to the robustness of the stability system. Short impulses, as used in the open loop analysis, had no appreciable effect on the aircraft states with the stability system engaged. Figure 27 is the time response of pitch angle, theta, and pitch rate to an impulse input.

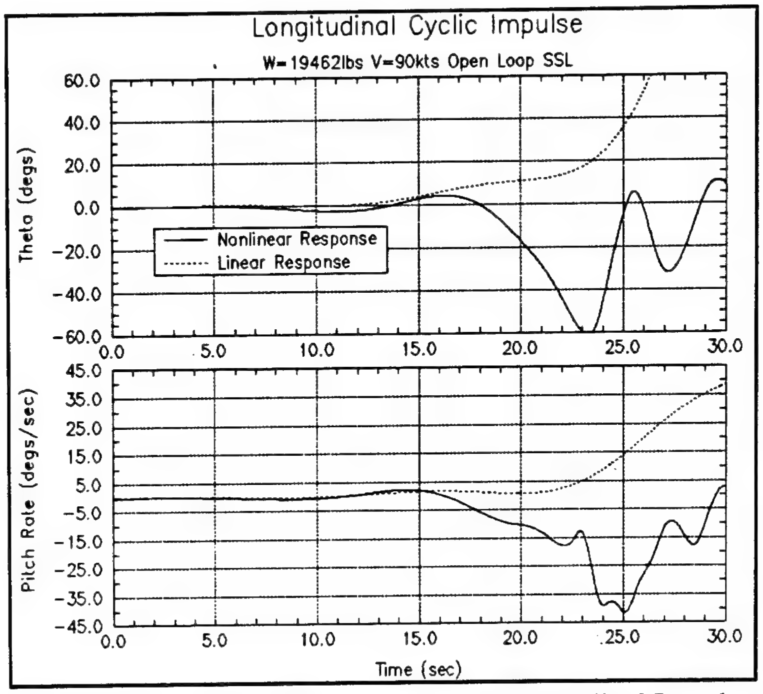


Figure 27. Open Loop Response From Longitudinal Impulse

The linear and nonlinear response are very close up to 15 seconds, after which they radically depart. The linear response diverges without oscillation, so it must be associated with the positive real eigenvalue.

The second test conducted was a lateral cyclic impulse also of one inch for a duration of .025 seconds. Figure 28 is the time history plot of roll angle, phi, and roll rate. The linear and nonlinear responses are very close up to about 20 seconds. After that the two depart, but show the same general trend up to 30 seconds. The linear response diverges and oscillates, so the lateral mode must be associated with the complex real eigenvalues. In both instances, the rapid departure of the linear and nonlinear responses was due to the aircraft states departing from the small linear range in which the linear model is valid.

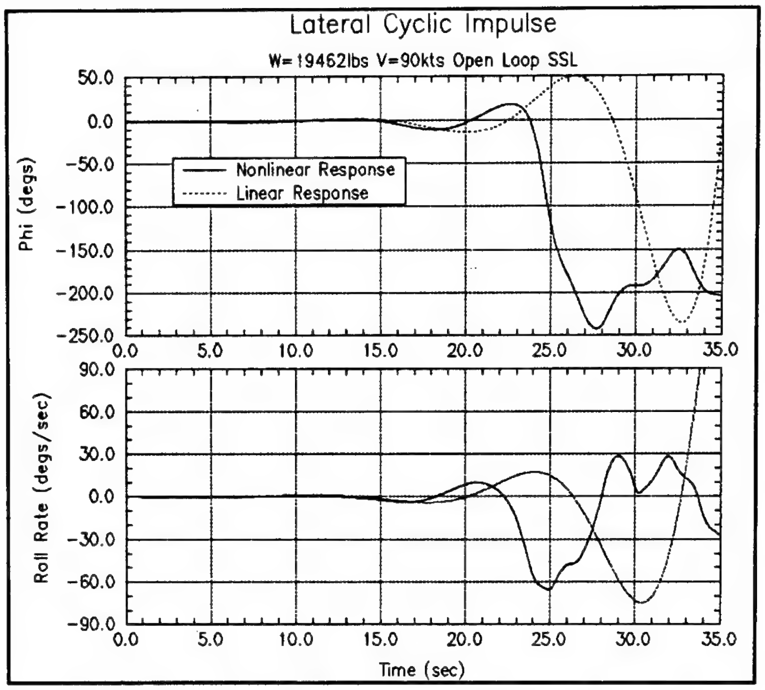


Figure 28. Open Loop Response From Lateral Impulse

Control input tests were also conducted for the collective and tail rotor impulse inputs, but the responses showed the same instability. This is expected as the strong cross coupling in the model, and in helicopters in general, causes a change in the lateral axis when an input is made into the collective, for instance. As already demonstrated, even the smallest perturbation caused the model to rapidly depart from the trimmed steady state flight condition.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

This thesis presents an outline on the use of Flightlab for rotorcraft modeling and analysis. A tutorial was presented to highlight the three programs within Flightlab, Model Editor, Gscope, and Model Analysis. The steps to implement a supercomponent, i.e. control system, into a model created under Model Editor were also covered. The same generic process can be followed for the insertion of any supercomponent that differs from the generalized model structures available in Model Editor.

The results of the SH-60B analysis compared favorably with the results of GenHel, Sikorsky Aircrafts modeling and analysis software. All of the comparisons were trimmed steady state flight, so the fact that the SAS channels were approximations had little or no effect on the results. The largest effect was in the closed loop analysis, but to what extent could not be assessed as no similar data was available for the full SAS system responses. The closed loop analysis showed that the model was stable as all of the eigenvalues of the $\tilde{A}$ matrix were in the left hand plane. It also demonstrated that the linear and nonlinear responses of the model to control inputs were very close. The open loop analysis showed that the helicopter was unstable in pitch, purely divergent, and in roll, oscillatory and divergent. The strong cross coupling between the six degrees of freedom of the linearized model was also shown.

B. RECOMMENDATIONS FOR FURTHER RESEARCH

The modeling and analysis conducted on the SH-60B yielded good results, but there are several improvements that could be made to enhance it's fidelity. First are the analog and digital SAS channels. Both systems implemented for this thesis are not the full scale systems, but approximations. Discussed in Chapter III, these systems are proprietary and could not be supplied by Sikorsky Aircraft. If though, through an agreement, the full systems could be supplied, they should be implemented. Providing the SAS channels are not too complex, modeling them in Flightlab would not be too much effort, and this thesis would be of great assistance in doing so.

Second, the main rotor was modeled as a rigid system. Flightlab has the capability to model it as elastic with up to ten mode shapes. The last two enhancements deal with the main rotor system aerodynamics. The induced velocity on the main rotor was modeled with the uniform inflow option, which is based on simple momentum theory. Either the free vortex or prescribed vortex selective modeling components can be used provided that the required data can be obtained. Finally, the airloads were modeled as quasi-steady, the dynamic stall option can be used, which allows for more accurate simulation results at and near stall conditions on the main rotor.

Another area for possible future work is the implementation of the aerodynamic data for the SH-60B in different mission configurations. Sikorsky Aircraft also supplied the data for lift, drag, pitching moment, and aerodynamic interference for torpedoes and external fuel tanks on the helicopter. The Model Editor does not contain options for these 'extra' aerodynamic panels and thus would need to be constructed in Gscope. The procedure would be similar to that outlined in this thesis for the control system.

Including the enhancements listed above would allow the model to be used for realistic engineering analysis of changes or improvements to the SH-60B. Suppression of high vibration levels is one possible area of research, which is an undesired trait of all helicopters and one that has received much attention. One technique for the suppression of rotorcraft vibrations is higher harmonic control (HHC), which is an active vibration system. One method of HHC is achieved by superimposing 4/rev swashplate motion upon basic collective and cyclic control inputs [Ref. 9: p. 3]. Reference 9 is a paper which presents the basic theory and results of a HHC system installed on a Army OH-6A helicopter. The paper also refers the reader to a plethora of other references for HHC theory. The main rotor is already modeled to accept swashplate inputs, as demonstrated, and the control system components are flexible enough to model the HHC system. Since HHC is designed to suppress fuselage vibration levels, the present modeling of the fuselage would need to be looked at in regard to accurately predicting vibration levels since it is a rigid model, and not elastic, which would be ideal.

Finally, the SH-60B model could be modified to enable real time engineering flight simulation for use with the pilots workstation. Flightlab requires the use of a multi-processor computer because three programs must be run simultaneously, the visual scene program, the model program, and the program to operate the pilots workstation [Ref. 2:p. 52]. Currently, the necessary hardware for piloted simulations is present in the computer center visualization laboratory at the Naval Postgraduate School.

LIST OF REFERENCES

1. *Aviation Week and Space Technology, LHX Design Proposals Spur Investment in Simulation Facilities,* 14 January 1985.

2. *Procedural Guide for Modelling and Analyzing the Flight Characteristics of a Helicopter Using Flightlab,* Gary P. McVaney, 23 September 1993.

3. *Flightlab/Scope Users Tutorial Manual,* Advanced Rotorcraft Technologies, 22 July 1993.

4. *Flightlab/Scope Component Reference Manual,* Advanced Rotorcraft Technologies, 22 July 1993.

5. *Flightlab/Scope Theory Manual,* Advanced Rotorcraft Technologies, 22 July 1993.

6. *Flightlab/Scope Command Reference Manual,* Advanced Rotorcraft Technologies, 22 July 1993.

7. *Flightlab Training Manual,* Advanced Rotorcraft Technologies.

8. *SH-60B Seahawk Simulation Model Data ETR-T1-068,* United Technologies Corporation, 17 May 1994.

9. *Journal of the American Helicopter Society, On Developing and Flight Testing a Higher Harmonic Control System,* E. Roberts Wood, Richard W. Powers, John H. Cline, C. Eugene Hammond, Vol. 30, No. 2, pp 3-20, January 1985.

APPENDIX A. FLIGHTLAB COMPONENT CLASS ICONS

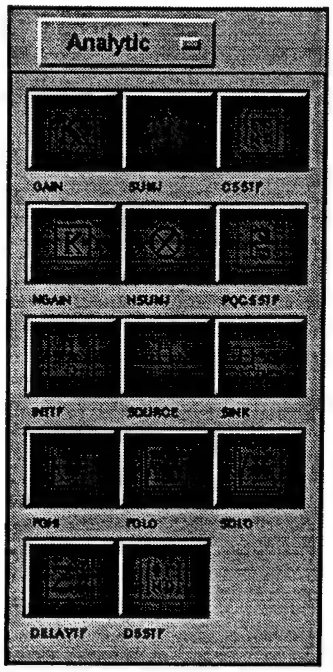


Figure 29. Aerodynamic Components

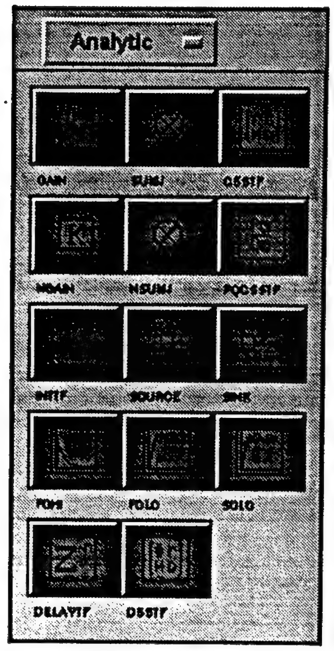


Figure 30. Analytic Components

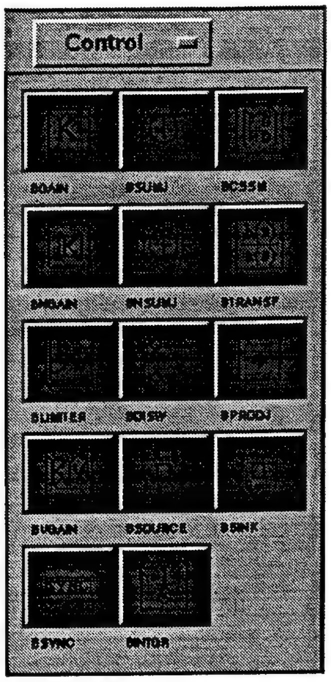


Figure 31. Control Components

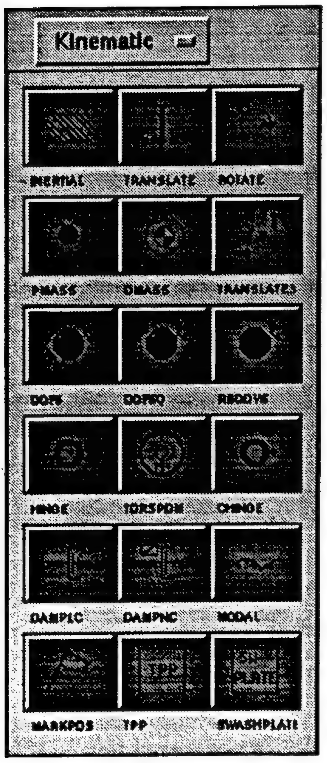


Figure 32. Kinematic Comonents

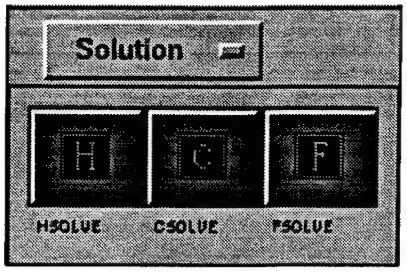


Figure 33. Solution Components

APPENDIX B. SH-60B DATA FILES

```
#Filename sh60fuseclal.sav
#user Wagner
#SEAHAWK FUSELAGE LIFT VS ALFWF LOW ANGLE MAP
#MAPNAME     =LQFMP
#MAPTYPE     =UVRUVR
#INPUT       =ALFWF
#OUTPUT      =LQF
CLFAL 13 1 0
-4.73700e+1 -3.91100e+1 -2.91000e+1 -2.16000e+1
-9.60000e00  0.95000e00  6.10000e00  9.95000e00
 1.42500e+1  1.90000e+1  2.57300e+1  3.17500e+1
 3.78700e+1

#Filename sh60fuseclah.sav
#user Wagner
#SEAHAWK FUSELAGE LIFT VS ALFWF HIGH ANGLE MAP
#MAPNAME     =LQFMP
#MAPTYPE     =UVRUVR
#INPUT       =ALFWF
#OUTPUT      =LQF
CLFAH 19 1 0
-2.10000e+1 -2.31000e+1 -3.21400e+1 -4.28600e+1
-5.04600e+1 -5.21600e+1 -4.73700e+1 -2.91000e+1
-9.60000e00  6.10000e00  1.42500e+1  2.57300e+1
 3.78700e+1  4.83900e+1  4.97100e+1  4.62100e+1
 4.06600e+1  3.26300e+1  2.20000e+1

#Filename sh60fusecmal.sav
#user Wagner
#SEAHAWK FUSELAGE PITCH MOMENT VS ALFWF LOW ANGLE MAP
#MAPNAME     =MQFMP
#MAPTYPE     =UVRUVR
#INPUT       =ALFWF
#OUTPUT      =MQF
CMFAL 13 1 0
-8.17360e+2 -7.20650e+2 -6.23500e+2 -6.09450e+2
-4.24390e+2 -2.97340e+2 -1.50760e+2 -3.73800e+1
 1.07500e+1  7.52000e+1  1.14110e+2  1.77610e+2
 2.40430e+2
```

```
#Filename sh60fusecmah.sav
#user Wagner
#SEAHAWK FUSELAGE PITCH MOMENT VS ALFWF HIGH ANGLE MAP
#MAPNAME    =MQFMP
#MAPTYPE    =UVRUVR
#INPUT      =ALFWF
#OUTPUT     =MQF
CMFAH 19 1 0
-1.32620e+2 -3.81080e+2 -5.97630e+2
-7.45910e+2 -8.31090e+2 -8.80820e+2
-8.17360e+2 -6.23500e+2 -4.24390e+2
-1.50760e+2  1.07500e+1  1.14110e+2
 2.40430e+2  2.95910e+2  2.71910e+2
 1.95290e+2  7.24100e+1 -9.92100e+1
-3.07380e+2

#Filename sh60fuseclbal.sav
#user Wagner
#SEAHAWK FUSELAGE DELTA LIFT VS PSIWF
#MAPNAME    =DLQFMP
#MAPTYPE    =BIV
#INPUT1     =PSIWF
#INPUT2     =ALFWF
#OUTPUT     =DLQF
CLFBAL 13 3 0
-1.75000e+1 -1.40000e+1 -9.40000e00 -4.60000e00
-1.70000e00 -2.10000e00  0.00000e00  0.20000e00
 0.70000e00  0.20000e00 -0.20000e00  1.40000e00
-3.60000e00 -3.00000e00 -1.30000e00  0.50000e00
 2.20000e00  2.90000e00 -0.80000e00  0.00000e00
-1.00000e00 -1.10000e00 -1.10000e00 -4.60000e00
-8.20000e00 -1.19000e+1  1.25000e+1  1.20000e+1
 9.50000e00  4.60000e00  4.80000e00  2.90000e00
 0.00000e00  1.20000e00  3.70000e00  5.20000e00
 6.20000e00  5.90000e00  4.90000e00
```

```
#Filename sh60fuseclbal.sav
#user Wagner
#SEAHAWK FUSELAGE DELTA LIFT VS PSIWF
#MAPNAME     =DLQFMP
#MAPTYPE     =BIV
#INPUT1      =PSIWF
#INPUT2      =ALFWF
#OUTPUT      =DLQF
CLFBAL 13 3 0
-1.75000e+1 -1.40000e+1 -9.40000e00 -4.60000e00
-1.70000e00 -2.10000e00  0.00000e00  0.20000e00
 0.70000e00  0.20000e00 -0.20000e00  1.40000e00
-3.60000e00 -3.00000e00 -1.30000e00  0.50000e00
 2.20000e00  2.90000e00 -0.80000e00  0.00000e00
-1.00000e00 -1.10000e00 -1.10000e00 -4.60000e00
-8.20000e00 -1.19000e+1  1.25000e+1  1.20000e+1
 9.50000e00  4.60000e00  4.80000e00  2.90000e00
 0.00000e00  1.20000e00  3.70000e00  5.20000e00
 6.20000e00  5.90000e00  4.90000e00

#Filename sh60fuseclbah.sav
#user Wagner
#SEAHAWK FUSELAGE DELTA LIFT VS PSIWF
# HIGH ANGLE MAP
#MAPNAME     =DLQFMP
#MAPTYPE     =BIV
#INPUT1      =PSIWF
#INPUT2      =ALFWF
#OUTPUT      =DLQF
CLFBAH 19 3 0
-1.75000e+1 -1.75000e+1 -1.75000e+1 -1.75000e+1
-1.75000e+1 -1.75000e+1 -1.75000e+1 -9.40000e+1
-1.70000e00  0.00000e00  7.00000e-1 -2.00000e-1
-3.60000e00 -3.60000e00 -3.60000e00 -3.60000e00
-3.60000e00 -3.60000e00 -3.60000e00 -3.00000e00
-3.00000e00 -3.00000e00 -3.00000e00 -3.00000e00
-3.00000e00 -3.00000e00  5.00000e-1  2.90000e00
 0.00000e00 -1.10000e00 -4.60000e00 -1.19000e+1
-1.19000e+1 -1.19000e+1 -1.19000e+1 -1.19000e+1
-1.19000e+1 -1.19000e+1  1.25000e+1  1.25000e+1
 1.25000e+1  1.25000e+1  1.25000e+1  1.25000e+1
 1.25000e+1  9.50000e00  4.80000e00  0.00000e00
 3.70000e00  6.20000e00  4.90000e00  4.90000e00
 4.90000e00  4.90000e00  4.90000e00  4.90000e00
 4.90000e00
```

```
#Filename sh60fusecdbl.sav
#user Wagner
#SEAHAWK FUSELAGE DELTA DRAG VS PSIWF LOW ANGLE MAP
#MAPNAME     =DDQFMP
#MAPTYPE     =BIVBIV
#INPUT1      =PSIWF
#INPUT2      =ALFWF
#OUTPUT      =DDQF
CDFBL 13 1 0
2.93000e+1
2.30000e+1
1.64000e+1
9.70000e00
4.60000e00
6.00000e-1
0.00000e00
1.00000e00
3.00000e00
8.00000e00
1.53000e+1
2.28000e+1
2.97000e+1
```

```
#Filename sh60fusecdbh.sav
#user Wagner
#SEAHAWK FUSELAGE DELTA DRAGVS PSIWF HIGH ANGLE MAP
#MAPNAME     =DDQFMP
#MAPTYPE     =BIVBIV
#INPUT1      =PSIWF
#INPUT2      =ALFWF
#OUTPUT      =DDQF
CDFBH 19 1 0
 1.71000e+2  1.68000e+2  1.62000e+2  1.41000e+2
 1.12000e+2  7.60000e+1  2.93000e+1  6.40000e00
4.60000e00   0.00000e00  3.00000e00  1.53000e+1
2.97000e+1   7.60000e+1  1.15000e+2  1.40000e+2
1.58000e+2   1.68000e+2  1.71000e+2
```

```
#Filename sh60fusecmbal.sav
#user Wagner
#SEAHAWK FUSELAGE DELTA PITCH MOM VS PSIWF
#MAPNAME     =DMQFMP
#MAPTYPE     =BIV
#INPUT1      =PSIWF
#INPUT2      =ALFWF
#OUTPUT      =DMQF
CMFBAL 13 3 0
 8.20000e+1  5.90000e+1  4.20000e+1
 2.60000e+1  2.00000e+1  1.80000e+1
 0.00000e00  0.00000e00 -7.00000e00
-2.00000e+1 -4.75000e+1 -8.70000e+1
-1.16000e+2 -8.00000e+1 -7.60000e+1
-7.00000e+1 -6.20000e+1 -3.90000e+1
-5.00000e00  0.00000e00 -1.60000e+1
-3.50000e+1 -8.00000e+1 -1.06500e+2
-1.23000e+2 -1.40000e+2 -8.00000e+1
-9.50000e+1 -1.00000e+2 -6.60000e+1
-5.60000e+1 -1.50000e+1  7.50000e+1
 1.20000e+2  1.71000e+2  1.86000e+2
 1.97000e+2  2.05000e+2  2.08000e+2

#Filename sh60fusecybal.sav
#user Wagner
#SEAHAWK FUSELAGE SIDE FORSE VS PSIWF LO ANGLE MAP
#MAPNAME     =YQFMP
#MAPTYPE     =UVRUVR
#INPUT1      =PSIWF
#INPUT2      =ALFWF
#OUTPUT      =YQF
CYFBAL 13 3 0
-7.20000e+1 -5.95000e+1 -4.70000e+1
-3.47500e+1 -2.20000e+1 -9.75000e00
 2.50000e00  1.30000e+1  2.35000e+1
 3.75000e+1  5.15000e+1  6.50000e+1
 7.85000e+1 -7.20000e+1 -5.95000e+1
-4.70000e+1 -3.47500e+1 -2.20000e+1
-9.75000e00  2.50000e00  1.30000e+1
 2.35000e+1  3.75000e+1  5.15000e+1
 6.50000e+1  7.85000e+1 -7.20000e+1
-5.95000e+1 -4.70000e+1 -3.47500e+1
-2.20000e+1 -9.75000e00  2.50000e00
 1.30000e+1  2.35000e+1  3.75000e+1
 5.15000e+1  6.50000e+1  7.85000e+1
```

```
#Filename sh60fusecybah.sav
#user Wagner
#SEAHAWK FUSELAGE SIDE FORCE VS PSIWF HIGH ANGLE MAP
#MAPNAME     =YQFMP
#MAPTYPE     =UVRUVR
#INPUT1      =PSIWF
#INPUT2      =PSIWF
#OUTPUT      =YQF
CYFBAH 19 3 0
-4.0000e+1  -6.60000e+1 -8.65000e+1
-9.6000e+1  -9.55000e+1 -8.70000e+1
-7.2000e+1  -4.70000e+1 -2.20000e+1
 2.5000e00   2.35000e+1  5.15000e+1
 7.8500e+1   9.40000e+1  9.95000e+1
 9.5000e+1   8.00000e+1  5.95000e+1
 3.9500e+1  -4.0000e+1  -6.60000e+1
-8.65000e+1 -9.6000e+1  -9.55000e+1
-8.70000e+1 -7.2000e+1  -4.70000e+1
-2.20000e+1  2.5000e00   2.35000e+1
 5.15000e+1  7.8500e+1   9.40000e+1
 9.95000e+1  9.5000e+1   8.00000e+1
 5.95000e+1  3.9500e+1   -4.0000e+1
-6.60000e+1 -8.65000e+1  -9.6000e+1
-9.55000e+1 -8.70000e+1  -7.2000e+1
-4.70000e+1 -2.20000e+1   2.5000e00
 2.35000e+1  5.15000e+1  7.8500e+1
 9.40000e+1  9.95000e+1  9.5000e+1
 8.00000e+1  5.95000e+1  3.9500e+1

#Filename sh60fusecnbal.sav
#user Wagner
#SEAHAWK FUSELAGE FOLLING MOMENT VS PSIWF
#MAPNAME     =RQFMP
#MAPTYPE     =UVRUVR
#INPUT       =PSIWF
#OUTPUT      =RQF
CNFBAL 13 3 0
1.25000e+1  1.80000e+1  2.00000e+1
1.80000e+1  1.20000e+1  1.10000e+1
1.25000e+1  4.00000e+1  7.20000e+1
9.80000e+1  1.08000e+2  1.11000e+2
1.16000e+2  1.25000e+1  1.80000e+1
2.00000e+1  1.80000e+1  1.20000e+1
1.10000e+1  1.25000e+1  4.00000e+1
7.20000e+1  9.80000e+1  1.08000e+2
1.11000e+2  1.16000e+2  1.25000e+1
1.80000e+1  2.00000e+1  1.80000e+1
1.20000e+1  1.10000e+1  1.25000e+1
4.00000e+1  7.20000e+1  9.80000e+1
1.08000e+2  1.11000e+2  1.16000e+2
```

```
#Filename sh60fusecnbah.sav
#user Wagner
#SEAHAWK FUSELAGE ROLLING MOMENT VS PSIWF HIGH ANGLE MAP
#MAPNAME     =RQFMP
#MAPTYPE     =UVRUVR
#INPUT       =PSIWF
#OUTPUT      =RQF
CNFBAH 19 3 0
-5.90000e+1 -5.10000e+1 -4.20000e+1
-3.10000e+1 -2.00000e+1 -7.00000e00
 1.25000e+1  2.00000e+1  1.20000e+1
 1.25000e+1  7.20000e+1  1.08000e+2
 1.16000e+2  1.18000e+2  1.19500e+2
 1.20000e+2  1.20500e+2  1.20750e+2
 1.21000e+2 -5.90000e+1 -5.10000e+1
-4.20000e+1 -3.10000e+1 -2.00000e+1
-7.00000e00  1.25000e+1  2.00000e+1
 1.20000e+1  1.25000e+1  7.20000e+1
 1.08000e+2  1.16000e+2  1.18000e+2
 1.19500e+2  1.20000e+2  1.20500e+2
 1.20750e+2  1.21000e+2 -5.90000e+1
-5.10000e+1 -4.20000e+1 -3.10000e+1
-2.00000e+1 -7.00000e00  1.25000e+1
 2.00000e+1  1.20000e+1  1.25000e+1
 7.20000e+1  1.08000e+2  1.16000e+2
 1.18000e+2  1.19500e+2  1.20000e+2
 1.20500e+2  1.20750e+2  1.21000e+2

#Filename sh60fusecrbal.sav
#user Wagner
#SEAHAWK FUSELAGE YAWING MOMENT VS PSIWF LOW ANGLE MAP
#MAPNAME     =BQFMP
#MAPTYPE     =BIVBIV
#INPUT1      =PSIWF
#INPUT2      =ALFWF
#OUTPUT      =NQF
CRFBAL 13 3 0
-1.30000e+2 -1.67000e+2  -1.96000e+2
-2.03000e+2 -1.56000e+2  -7.80000e+1
 3.50000e+1  1.25000e+2   2.00000e+2
 2.35000e+2  2.46000e+2   2.39000e+2
 2.29000e+2 -9.25000e+1  -1.27000e+2
-1.56000e+2 -1.75000e+2  -1.34000e+2
 2.00000e+1  7.30000e+1   1.48000e+2
 2.42000e+2  2.76000e+2   2.88000e+2
 2.80000e+2  2.74000e+2  -8.00000e+1
-9.50000e+1 -1.00000e+2  -6.60000e+1
-5.60000e+1 -1.5000e+1    7.50000e+1
 1.20000e+2  1.71000e+2   1.86000e+2
 1.97000e+2  2.05000e+2   2.08000e+2
```

```
#Filename sh60fusecrbah.sav
#user Wagner
#SEAHAWK FUSELAGE YAWING MOMENT VS PSIWF HIGH ANGLE MAP
#MAPNAME    =NQFMP
#MAPTYPE    =BIVBIV
#INPUT1     =PSIWF
#INPUT2     =ALFWF
#OUTPUT     =NQF
CRFBAH 19 3 0
 4.40000e+2  3.92000e+2  3.32000e+2
 2.59000e+2  1.60000e+2  4.00000e+1
-1.30000e+2 -1.96000e+2 -1.56000e+2
 3.50000e+1  2.00000e+2  2.46000e+2
 2.29000e+2  2.45000e+2  1.80000e+2
 8.80000e+1 -4.00000e+1 -2.08000e+2
-4.20000e+2  4.40000e+2  3.92000e+2
 3.32000e+2  2.59000e+2  1.60000e+2
 4.00000e+1 -9.25000e+1 -1.56000e+2
-1.34000e+2  7.30000e+1  2.42000e+2
 2.88000e+2  2.74000e+2  2.45000e+2
 1.80000e+2  8.80000e+1 -4.00000e+1
-2.08000e+2 -4.20000e+2  4.40000e+2
 3.92000e+2  3.32000e+2  2.59000e+2
 1.60000e+2  4.00000e+1 -8.00000e+1
-1.00000e+2 -5.60000e+1  7.50000e+1
 1.71000e+2  1.97000e+2  2.08000e+2
 2.45000e+2  1.80000e+2  8.80000e+1
-4.00000e+1 -2.08000e+2 -4.20000e+2
```

```
#Filename sh60fusecmbah.sav
#user Wagner
#SEAHAWK FUSELAGE DELTA PITCH MOM VS PSIWF
#HI ANGLE MAP
#MAPNAME     =DMQFMP
#MAPTYPE     =BIV
#INPUT1      =PSIWF
#INPUT2      =ALFWF
#OUTPUT      =DMQF
CMFBAH 19 3 0
 8.20000e+1  8.20000e+1  8.20000e+1
 8.20000e+1  8.20000e+1  8.20000e+1
 8.20000e+1  4.20000e+1  2.00000e+1
 0.00000e00 -7.00000e00 -4.75000e+1
-1.16000e+2 -1.16000e+2 -1.16000e+2
-1.16000e+2 -1.16000e+2 -1.16000e+2
-1.16000e+2 -8.00000e+1 -8.00000e+1
-8.00000e+1 -8.00000e+1 -8.00000e+1
-8.00000e+1 -8.00000e+1 -7.00000e+1
-3.90000e+1 0.00000e00  -3.50000e+1
-1.06500e+2 -1.40000e+2 -1.40000e+2
-1.40000e+2 -1.40000e+2 -1.40000e+2
-1.40000e+2 -1.40000e+2 -8.00000e+1
-8.00000e+1 -8.00000e+1 -8.00000e+1
-8.00000e+1 -8.00000e+1 -8.00000e+1
-1.00000e+2 -5.60000e+1  7.50000e+1
 1.71000e+2  1.97000e+2  2.08000e+2
 2.08000e+2  2.08000e+2  2.08000e+2
 2.08000e+2  2.08000e+2  2.08000e+2
#Filename sh60fusecmbal.sav
#user Wagner
#SEAHAWK FUSELAGE DELTA PITCH MOM VS PSIWF
#MAPNAME     =DMQFMP
#MAPTYPE     =BIV
#INPUT1      =PSIWF
#INPUT2      =ALFWF
#OUTPUT      =DMQF
CMFBAL 13 3 0
 8.20000e+1  5.90000e+1  4.20000e+1
 2.60000e+1  2.00000e+1  1.80000e+1
 0.00000e00  0.00000e00 -7.00000e00
-2.00000e+1 -4.75000e+1 -8.70000e+1
-1.16000e+2 -8.00000e+1 -7.60000e+1
-7.00000e+1 -6.20000e+1 -3.90000e+1
-5.00000e00  0.00000e00 -1.60000e+1
-3.50000e+1 -8.00000e+1 -1.06500e+2
-1.23000e+2 -1.40000e+2 -8.00000e+1
-9.50000e+1 -1.00000e+2 -6.60000e+1
-5.60000e+1 -1.50000e+1  7.50000e+1
 1.20000e+2  1.71000e+2  1.86000e+2
 1.97000e+2  2.05000e+2  2.08000e+2
```

```
#Filename sh60fusecybal.sav
#user Wagner
#SEAHAWK FUSELAGE SIDE FORSE VS PSIWF LO ANGLE MAP
#MAPNAME    =YQFMP
#MAPTYPE    =UVRUVR
#INPUT1     =PSIWF
#INPUT2     =ALFWF
#OUTPUT     =YQF
CYFBAL 13 3 0
-7.20000e+1 -5.95000e+1 -4.70000e+1
-3.47500e+1 -2.20000e+1 -9.75000e00
 2.50000e00  1.30000e+1  2.35000e+1
 3.75000e+1  5.15000e+1  6.50000e+1
 7.85000e+1 -7.20000e+1 -5.95000e+1
-4.70000e+1 -3.47500e+1 -2.20000e+1
-9.75000e00  2.50000e00  1.30000e+1
 2.35000e+1  3.75000e+1  5.15000e+1
 6.50000e+1  7.85000e+1 -7.20000e+1
-5.95000e+1 -4.70000e+1 -3.47500e+1
-2.20000e+1 -9.75000e00  2.50000e00
 1.30000e+1  2.35000e+1  3.75000e+1
 5.15000e+1  6.50000e+1  7.85000e+1

#Filename sh60fusecybah.sav
#user Wagner
#SEAHAWK FUSELAGE SIDE FORCE VS PSIWF HIGH ANGLE MAP
#MAPNAME    =YQFMP
#MAPTYPE    =UVRUVR
#INPUT1           =PSIWF
#INPUT2           =PSIWF
#OUTPUT           =YQF
CYFBAH 19 3 0
-4.0000e+1  -6.60000e+1 -8.65000e+1
-9.6000e+1  -9.55000e+1 -8.70000e+1
-7.2000e+1  -4.70000e+1 -2.20000e+1
 2.5000e00   2.35000e+1  5.15000e+1
 7.8500e+1   9.40000e+1  9.95000e+1
 9.5000e+1   8.00000e+1  5.95000e+1
 3.9500e+1  -4.0000e+1  -6.60000e+1
-8.65000e+1 -9.6000e+1  -9.55000e+1
-8.70000e+1 -7.2000e+1  -4.70000e+1
-2.20000e+1  2.5000e00   2.35000e+1
 5.15000e+1  7.8500e+1   9.40000e+1
 9.95000e+1  9.5000e+1   8.00000e+1
 5.95000e+1  3.9500e+1  -4.0000e+1
-6.60000e+1 -8.65000e+1 -9.6000e+1
-9.55000e+1 -8.70000e+1 -7.2000e+1
-4.70000e+1 -2.20000e+1  2.5000e00
 2.35000e+1  5.15000e+1  7.8500e+1
 9.40000e+1  9.95000e+1  9.5000e+1
 8.00000e+1  5.95000e+1  3.9500e+1
```

```
#Filename sh60htailcll.sav
#user Wagner
#SEAHAWK HORIZONTAL STAB LIFT COEFFICIENT VS ALFPP1
# (LOW ANGLE MAP -30 30 DELTA 5)
#MAPNAME     =CLP1MR
#MAPTYPE     =UVSUVS
#INPUT       =ALFPP1
#OUPUT       =CLP1
CLHL 13 1 0
-7.45000e-1 -7.95000e-1 -9.50000e-1 -1.01300e00
-7.45000e-1 -3.80000e-1  0.00000e00  3.80000e-1
 7.45000e-1  1.01300e00  9.50000e-1  7.95000e-1
 7.45000e-1

#Filename sh60htailclh.sav
#user Wagner
#SEAHAWK HORIZONTAL STAB LIFT COEFFICIENT VS ALFPP1
# (HIGH ANGLE MAP -90 90 DELTA 10)
#MAPNAME     =CLP1MR
#MAPTYPE     =UVSUVS
#INPUT       =ALFPP1
#OUPUT       =CLP1
CLHH  19 1 0
 0.00000e00 -1.50000e-1 -3.00000e-1 -4.40000e-1
-5.60000e-1 -6.50000e-1 -7.45000e-1 -9.50000e-1
-7.45000e-1  0.00000e00  7.45000e-1  9.50000e-1
 7.45000e-1  6.50000e-1  5.60000e-1  4.40000e-1
 3.00000e-1  1.50000e-1  0.00000e00

#Filename sh60vtailcdl.sav
#user Wagner
#SEAHAWK VERTICAL STABILZER DRAG COEFF VS ALFPP3
# (LOW ANGLE MAP -30 30 DELTA 5)
#MAPNAME     =CDP3MP
#MAPTYPE     =UVRUVR
#INPUT       =ALFPP3
#OUTPUT      =CDP3
CDVL 13 1 0
3.60000e-1  2.65000e-1  1.74000e-1  1.18000e-1
6.60000e-2  3.30000e-2  1.80000e-2  2.10000e-2
4.40000e-2  9.20000e-2  1.62000e-1  2.48000e-1
3.55000e-1
```

```
#Filename sh60htailclh.sav
#user Wagner
#SEAHAWK HORIZONTAL STAB LIFT COEFFICIENT VS ALFPP1
# (HIGH ANGLE MAP -90 90 DELTA 10)
#MAPNAME    =CLP1MR
#MAPTYPE    =UVSUVS
#INPUT      =ALFPP1
#OUPUT      =CLP1
CLHH  19 1 0
 0.00000e00 -1.50000e-1 -3.00000e-1 -4.40000e-1
-5.60000e-1 -6.50000e-1 -7.45000e-1 -9.50000e-1
-7.45000e-1  0.00000e00  7.45000e-1  9.50000e-1
 7.45000e-1  6.50000e-1  5.60000e-1  4.40000e-1
 3.00000e-1  1.50000e-1  0.00000e00

#Filename sh60vtailcdl.sav
#user Wagner
#SEAHAWK VERTICAL STABILZER DRAG COEFF VS ALFPP3
# (LOW ANGLE MAP -30 30 DELTA 5)
#MAPNAME    =CDP3MP
#MAPTYPE    =UVRUVR
#INPUT      =ALFPP3
#OUTPUT     =CDP3
CDVL 13 1 0
3.60000e-1  2.65000e-1  1.74000e-1  1.18000e-1
6.60000e-2  3.30000e-2  1.80000e-2  2.10000e-2
4.40000e-2  9.20000e-2  1.62000e-1  2.48000e-1
3.55000e-1

#Filename sh60vtailcdh.sav
#user Wagner
#SEAHAWK VERTICAL STABILZER DRAG COEFF VS ALFPP3
# (HIGH ANGLE MAP -90 90 DELTA 10)
#MAPNAME    =CDP3MP
#MAPTYPE    =UVRUVR
#INPUT      =ALFPP3
#OUTPUT     =CDP3
CDVH 19 1 0
1.10000e00  1.02500e00  9.65000e-1  8.75000e-1
7.45000e-1  5.75000e-1  3.60000e-1  1.74000e-1
6.60000e-2  1.80000e-2  4.40000e-2  1.62000e-1
3.55000e-1  5.80000e-1  7.50000e-1  8.75000e-1
9.65000e-1  1.02000e00  1.08000e00
```

```
#Filename sh60vtailcll.sav
#user Wagner
#SEAHAWK VERTICAL STABILZER LIFT COEFF VS ALFPP3
# (LOW ANGLE MAP -30 30 DELTA 5)
#MAPNAME     =CLP3MP
#MAPTYPE     =UVRUVR
#INPUT       =ALFPP3
#OUTPUT      =CLP3
CLVL 13 1 0
-1.00000e00 -1.00000e00 -9.30000e-1 -7.30000e-1
-5.00000e-1 -2.80000e-1 -6.00000e-2  1.60000e-1
 3.80000e-1  6.10000e-1  8.20000e-1  8.90000e-1
 8.90000e-1

#Filename sh60vtailclh.sav
#user Wagner
#SEAHAWK VERTICAL STABILZER LIFT COEFF VS ALFPP3
# (HIGH ANGLE MAP -90 90 DELTA 10)
#MAPNAME     =CLP3MP
#MAPTYPE     =UVRUVR
#INPUT       =ALFPP3
#OUTPUT      =CLP3
CLVH 19 1 0
 0.00000e00 -1.20000e-1 -2.80000e-1 -4.60000e-1
-6.60000e-1 -8.80000e-1 -1.00000e00 -9.30000e-1
-5.00000e-1 -6.00000e-2  3.80000e-1  8.20000e-1
 8.900000e-1 8.00000e-1  6.30000e-1  4.80000e-1
 3.20000e-1  1.70000e-1  0.00000e00

#Filename mrint_fusex.sav
#user Wagner
#SEAHAWK FORE/AFT M.R. DOWNWASH AT FUSELAGE
#MAPNAME     =EXWFMP
#MAPTYPE     =BIV
#INPUT1      =CHIPMR
#INPUT2      =AA1FMR
#OUTPUT      =EKXWF
RVXFUSEI 11 3 0
8.00000e-2  1.80000e-1  3.00000e-1  4.30000e-1
5.50000e-1  6.60000e-1  7.90000e-1  9.00000e-1
1.03000e00  5.50000e-1  0.00000e00  0.00000e00
1.00000e-1  2.10000e-1  3.20000e-1  4.20000e-1
5.40000e-1  6.60000e-1  8.00000e-1  9.40000e-1
5.00000e-1  0.00000e00  -1.20000e-1 2.00000e-2
8.00000e-2  1.80000e-1  2.80000e-1  4.00000e-1
5.30000e-1  6.70000e-1  8.20000e-1  4.00000e-1
0.00000e00
```

```
#Filename mrint_fusey.sav
#user Wagner
#SEAHAWK LATERAL M.R. DOWNWASH AT FUSELAGE
#MAPNAME     =EYWFMP
#INPUT1      =CHIPMR
#INPUT2      =AA1FMR
#OUTPUT      =EKZWF
RVYFUSEI 11 3 0
0.0    0.0    0.0    0.0
0.0    0.0    0.0    0.0
0.0    0.0    0.0    0.0
0.0    0.0    0.0    0.0
0.0    0.0    0.0    0.0
0.0    0.0    0.0    0.0
0.0    0.0    0.0    0.0
0.0    0.0    0.0    0.0
0.0

#Filename mrint_fusez.sav
#user Wagner
#SEAHAWH VERTICAL M.R. DOWNWASH AT FUSELAGE
#  Sign is opposite that of Sikorsy ETR-T1-068 due to
#  difference in coordinate system orientation.  Flightlab
#  has x forward y out right wing z down where Sikorsky data
#  has x forward y out left wing and z up.
#MAPNAME     =EZWFMP
#MAPTYPE     =BIV
#INPUT1      =CHIPMR
#INPUT2      =EKZWF
#OUTPUT      =EKZWF
RVZFUSEI 11 3 0
-1.11000e00 -1.09000e00 -1.08000e00 -1.06500e00
-1.05000e00 -1.04000e00 -1.02000e00 -1.01000e00
-1.00000e00 -8.80000e-1 -6.00000e-1 -1.12000e00
-1.12000e00 -1.12000e00 -1.12000e00 -1.12000e00
-1.12000e00 -1.12000e00 -1.12000e00 -1.11000e00
-9.60000e-1 -6.00000e-1 -1.15000e00 -1.15000e00
-1.15000e00 -1.15000e00 -1.16000e00 -1.17000e00
-1.18000e00 -1.22000e00 -1.16000e00 -9.80000e-1
-6.00000e-1
```

```
#Filename mrint_htailx.sav
#user Wagner
#SEAHAWK FORE/AFT M.R. DOWNWASH AT HORIZONTAL TAIL
#MAPNAME     =EXP1MP
#MAPTYPE     =BIV
#INPUT1      =CHIPMR
#INPUT2      =AA1FMR
#OUTPUT      =EKXP1
RVXHTI 11 3 0
 0.00000e00 -2.00000e-1  5.00000e-2
 3.00000e-1  5.40000e-1  8.00000e-1
 1.04000e00  1.30000e00  1.55000e00
 8.00000e-1  0.00000e00 -4.00000e-1
-6.00000e-1 -2.00000e-1  1.20000e-1
 3.60000e-1  6.00000e-1  8.30000e-1
 1.06000e00  1.30000e00  6.60000e-1
 0.00000e00 -5.60000e-1 -8.00000e-1
-7.40000e-1 -3.20000e-1  4.00000e-2
 3.20000e-1  6.00000e-1  8.60000e-1
 1.12000e00  5.40000e-1  0.00000e00
```

```
#Filename mrint_htaily.sav
#User Wagner
#SEAHAWK LATERAL M.R. DOWNWASH AT HORIZONTAL TAIL
#     RVYHTI
RVYHTI 11 3 0
  0.0000e+00   0.0000e+00   0.0000e+00
  0.0000e+00   0.0000e+00   0.0000e+00
  0.0000e+00   0.0000e+00   0.0000e+00
  0.0000e+00   0.0000e+00   0.0000e+00
  0.0000e+00   0.0000e+00   0.0000e+00
  0.0000e+00   0.0000e+00   0.0000e+00
  0.0000e+00   0.0000e+00   0.0000e+00
  0.0000e+00   0.0000e+00   0.0000e+00
  0.0000e+00   0.0000e+00   0.0000e+00
  0.0000e+00   0.0000e+00   0.0000e+00
  0.0000e+00   0.0000e+00   0.0000e+00
```

```
#Filename  mrint_htailz.sav
#User Wagner
#SEAHAWK VERTICAL M.R. DOWNWASH AT HORIZONTAL TAIL
#MAPNAME     =EZP1MP
#MAPTYPE     =BIV
#INPUT1      =CHIPMR
#INPUT2      =AA1FMR
#OUTPUT      =EKZP1
#     RVZHTI
RVZHTI 11 3 0
 1.30000e-1  -8.00000e-1  -1.80000e00
 -1.82000e00  -1.86000e00  -1.88000e00
 -1.91000e00  -1.94000e00  -1.69000e00
 -1.42000e00  -1.14000e00  -4.00000e-1
 -9.40000e-1  -1.84000e00  -1.91000e00
 -1.98000e00  -2.04000e00  -2.08000e00
 -2.14000e00  -1.89000e00  -1.62000e00
 -1.35000e00  -7.80000e-1  -1.36000e00
 -1.91000e00  -1.98000e00  -2.06000e00
 -2.14000e00  -2.21000e00  -2.28000e00
 -2.16000e00  -1.96000e00  -1.56000e00

#Filename  fuseint_stabqdyn.sav
#User Wagner
#SEAHAWK FDYNAMIC PRESSURE RATIO AT STABILATRO VS ALFWF
#MAPNAME     =QP1MP
#MAPTYPE     =UVR
#INPUT       =ALFWF
#OUTPUT      =QP1QWF
#
HTQDYN 1 13 0
1.00000e00  1.00000e00  9.50000e-1  8.00000e-1
6.65000e-1  6.50000e-1  7.50000e-1  8.15000e-1
7.30000e-1  7.60000e-1  9.10000e-1  9.80000e-1
1.00000e00
```

```
#Filename fuseintf_stabvy.sav
#User Wagner
#SEAHAWK FUSELAGE LATERAL DOWNWASH ON STABILATOR VS ALFWF
#     FVYHT
FVYHT 1 1 0
0.00000e00
#Filename  fuseintf_stabvz.sav
#User Wagner
#SEAHAWK FUSELAGE DOWNWASH ON STABILATOR VS ALFWF
#MAPNAME    =EPP1MP
#MAPTYPE    =UVRUVR
#INPUT      =ALFWF
#OUTPUT     =EPSSP1
#
FVZHT 1 13 0
 4.10000e00  3.50000e00  2.90000e00  1.90000e00
-2.50000e-1 -1.50000e00 -1.65000e00 -1.40000e00
-1.95000e00 -4.40000e00 -5.75000e00 -6.65000e00
-7.50000e00

#Filename fuseintf_vertqdyn.sav
#User Wagner
#SEAHAWK DYNAMIC PRESSURE RATIO AT VERITCAL TAIL VS PSIWF
#MAPNAME    =QP3MP
#MAPTYPE    =BIV
#INPUT1     =PSIWF
#INPUT2     =ALFWF
#OUTPUT     =QP3QWF
VTQDYN 15 3 0
1.00000e00  9.75000e-1  9.25000e-1  8.45000e-1
7.80000e-1  7.20000e-1  6.90000e-1  6.55000e-1
6.50000e-1  7.60000e-1  7.95000e-1  8.10000e-1
8.40000e-1  9.20000e-1  1.00000e00  1.00000e01
9.55000e-1  8.25000e-1  7.30000e-1  6.60000e-1
5.90000e-1  5.80000e-1  5.70000e-1  5.80000e-1
6.10000e-1  6.60000e-1  7.10000e-1  7.80000e-1
9.10000e-1  1.00000e01  1.00000e00  9.20000e-1
7.85000e-1  6.90000e-1  6.15000e-1  5.75000e-1
5.40000e-1  5.10000e-1  5.20000e-1  5.30000e-1
5.60000e-1  6.30000e-1  7.30000e-1  8.40000e-1
1.00000e00
```

```
#Filename fuseintf_vertvy.sav
#User Wagner
#SEAHAWK FUSELAGE SIDEWASH ON VERTICAL TAIL VS PSIWF
#  Sign is opposite that of Sikorksy ETR-T1-068 due to
#  difference in coordinate system orientation.  Flightlab
#  has x forward y out right wing z down where Sikorsky data
#  has x forward y out left wing and z up.
#MAPNAME    =SGP3MP
#MAPTYPE    =BIVBIV
#INPUT1     =PSIWF
#INPUT2     =ALFWF
#OUTPUT     =SIGP3
FVYVT 13 3 0
-7.80000e00 -7.00000e00 -5.50000e00 -1.00000e00
-5.00000e-1  4.00000e-1  1.75000e00  1.90000e00
 2.00000e00  2.20000e00  4.10000e00  5.30000e00
 5.80000e00  4.35000e00  3.35000e00  2.70000e00
 1.30000e00  1.70000e00  2.80000e00  1.55000e00
 2.00000e-1 -1.50000e00 -1.90000e00 -2.65000e00
-3.55000e00 -4.20000e00  4.40000e00  4.50000e00
5.00000e00   5.40000e00  5.20000e00  3.30000e00
2.20000e00  -9.00000e-1 -4.00000e00 -4.10000e00
-5.20000e00 -6.00000e00 -6.40000e00
```

```
# ///////////////////////////////////////////////////////////
# / file stab.sav
# / file location d4/ wagner/flightscope/models/table/
# /   rotorcraft
# / date 27 July 1995
# /
# / This file is the stabilator position map
# /   input 1 is forward airspeed (VXLAG)
# /   input 2 is collective position in percent (XCPC)
# ///////////////////////////////////////////////////////////
#
#
STABPOSIT 1 209 0
 42.36 42.36 42.36 42.36 32.40 22.40 12.47 5.02 -2.35
 -3.86 -5.00 -6.14 -7.28      -8.43 -9.57 -10.46 -10.46 -10.46
 -10.46 42.36 42.36     42.36 42.36 32.92 23.48 14.03 6.59
 -0.78 -2.29 -3.43 -4.57 -5.71 -6.86 -8.00 -8.89 -8.89
 -8.89 -8.89 42.36 42.36      42.36 42.36 33.44 24.51 15.59
 8.17 0.80 -.71 -1.86 -3.00 -4.19 -5.38 -6.43 -7.32
-7.32 -7.32 -7.32 42.36 42.36 42.36 42.36 33.96 25.55
17.14 9.74 2.37 .86     -.29  -1.43 -2.57 -3.72 -4.86
-5.75 -5.75 -5.75 -5.75 42.36 42.36 42.36 42.36 34.47
26.59 18.70 11.37 3.94 2.43 1.29 .14 1.00 -2.14
-3.29 -4.18 -4.18 -4.18 -4.18 42.36 42.36 42.36 42.36
34.99 27.62 20.25 12.88 5.51 4.00 2.86 1.72 .57
-.57  -1.71 -2.60 -2.60 -2.60 -2.60 42.36 42.36 42.36
42.36 36.40 30.43 24.46 17.17 9.77  5.57  4.43  3.29
2.14  1.00  -.14  -1.03 -1.03 -1.03 -1.03 42.36 42.36
42.36 42.36 37.80 33.24 28.67 21.39 14.12 7.15 6.00
4.86  3.71  2.57  1.43  .54   .54   .54   .54   42.36
42.36 42.36 42.36 37.80 33.24 28.67 21.39 14.12 7.15
6.00  4.86  3.71  2.57  1.43  .54 .54 .54 .54
42.36 42.36 42.36 42.36 37.80 33.24 28.67 21.39 14.12
7.15  6.00  4.86  3.71  2.57  1.43  .54   .54   .54
.54   42.36 42.36 42.36 42.36 37.80 33.24 28.67 21.39
14.12 7.15  6.00  4.86  3.71  2.57  1.43  .54   .54
.54   .54
```

APPENDIX C. MODEL SCRIPT FILES

```
//
// filename sh60b_rigid.def
// This file is created by Model Editor
//
clearm
describe("Model-Id: sh60b_rigid");
describe("Model-Description: SH60B Seahawk");
exec("$FL_SCRIPTS/startup.scr",1);

// First pass instructions: loading data

goto world;  // world

path("$FL_SCRIPTS/rotorcraft/");
pushg (model)

pushg (data)
isRotorcraft = 1;
popg // data

pushg (data)
//******* begin loading data *******
testflag = 0 "flag for test: 0: free fly; 1: wind tunnel test";
massflag = 0 "flag about the input of this screen: 1: total vehicle; 0:
fuselage only";
vmass  =  573.677  "vehicle  mass  (massflag=1:  total  vehicle  mass;   0:
fuselage only)";
fscg = 29.508 "Fuselage station of vehicle cg position (1)";
blcg = .0167 "Buttline station of vehicle cg position (1)";
wlcg = 20.7583 "Waterline station of vehicle cg position (1)";
Ixx = 6135 "X-axis (roll) moment of inertia (massflag=1: total vehicle; 0:
fuselage only)";
Iyy = 48715 "Y-axis (pitch) moment of inertia (massflag=1: total vehicle;
0: fuselage)";
Izz = 46785 "Z-axis (yaw) moment of inertia (massflag=1: total vehicle; 0:
fuselage)";
Ixy = -317 "X-Y -axis product of inertia (massflag=1: total vehicle; 0:
fuselage)";
Ixz = 2535 "X-Z axis product of inertia (massflag=1: total vehicle; 0:
fuselage)";
Iyz = -150 "Y-Z-axis product of inertia (massflag=1: total vehicle; 0:
fuselage)";
//******* end loading data *******
popg // data

exec("applpath.scr",1); // application path
path("~/sh60")

pushg (data)
```

```
      exec("rotorcraft.prolog",1);
popg // data
      exec("rotorcraft.exc",1);

path([])

exec("$FL_SCRIPTS/permpath.scr",1);

path("$FL_SCRIPTS/rotorcraft/environ_atmos/");
pushg (model)

pushg (data)
isEnvironAtmos = 1;
popg // data

pushg (data)
//******* begin loading data *******
//******* end loading data *******
popg // data

path("$FL_SCRIPTS/rotorcraft/environ_atmos/");
pushg (model)

pushg (data)
isAtmosphere = 1;
popg // data

pushg (data)
//******* begin loading data *******
atmtab = read("$FL_LOCAL_DB/tables/rotorcraft//atmo.tab");
describe(atmtab,"Atmosphere table, enter a filename, data is expected in
scope read format");
//******* end loading data *******
popg // data

popg // model
exec("applpath.scr",1); // application path
pushg (data)
      exec("environ_atmos.prolog",1);
popg // data
      exec("environ_atmos.exc",1);
popg // model

path([])

exec("$FL_SCRIPTS/permpath.scr",1);

path("$FL_SCRIPTS/rotorcraft/rotors/");
pushg (rotor1_rotor)

pushg (data)
```

```
isRotor1 = 1;
popg // data

pushg (data)
//******* begin loading data *******
fsmr = 28.4346 "Fuselage station of Rotor1 (l)";
blmr = 0.0 "Buttline  station of Rotor1 in inches";
wlmr = 26.25 "Waterline station of Rotor1 (l)";
nblades = 4 "Number of Rotor1 blades";
Btl = 0.97 "Rotor1 blade tip loss distance";
eulerhub = [0.0  177.0  0.0];
tiltaxis = 2 "Axis about which rotor shaft tilts as in tilt rotor. 1:
x-axis; 2: y; 3: z";
rpmnom = 27.0 "Rotor1 nominal speed in radian/sec";
rmr = 26.83 "Rotor1 radius in feet";
naz = 24 "Number of azimuth step/rev";
phase = -9.7 "swash plate phase angle (deg)";
//******* end loading data *******
popg // data

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/");
pushg (rotor)

pushg (data)
isBladeElement = 1;
popg // data

pushg (data)
//******* begin loading data *******
clockwise = 0 "0: counter-clockwise rotor; 1: clockwise rotor";
//******* end loading data *******
popg // data

popg // rotor

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/hub/");
pushg (rotor)

pushg (data)
isHub = 1;
popg // data

pushg (data)
//******* begin loading data *******
coneang = 0.0 "rotor precone angle in degree";
coneof = 0.0 "rotor precone offset from center of rotation in ft";
transylen = 0.0 "torque  offset in y-direction positive forward";
//******* end loading data *******
popg // data

popg // rotor
```

```
path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/hub/articulated/");
pushg (rotor)

pushg (data)
isArticulated = 1;
popg // data

pushg (data)
//******* begin loading data *******
laghinge = 1 "option for lag hinge: 0: no lag hinge; 1: with lag hinge";
flaphof = 1.25 "Rotor flap hinge offset in feet";
laghof = 1.25 "Rotor lag hinge offset in feet";
feathof = 1.25 "Rotor feathring hinge offset in feet";
delta3 = 0 "possitive delta3  for negative pitch-flap coupling";
flapkk = 0 "flap hinge rotational spring stiffness";
flapcc = 0 "flap hinge damping coefficient";
flapang0 = 0 "flap hinge spring undeformed angle";
//******* end loading data *******
popg // data

popg // rotor

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/hub/articulated/lagdam
per/linear/");
pushg (rotor)

pushg (data)
isLineardamper = 1;
popg // data

pushg (data)
//******* begin loading data *******
lagkk = 0 "linear lag damper stiffness";
lagcc = 20000 "linear lag damper damping coefficient";
lagang0 = 0 "linear lag damper spring undeformed angle";
//******* end loading data *******
popg // data

popg // rotor

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/blade/");
pushg (rotor)

pushg (data)
isBlade = 1;
popg // data

pushg (data)
//******* begin loading data *******
firstscpib = 1.25 "Inboard edge location of blade structure, typically
locates at outmost hinge";
```

```
scpoption = 1 "0: for equivalent mass inertia; 1: for equal annulii area";
bcgoff = read("$FL_LOCAL_DB/tables/rotorcraft//blade_cgoff.tab");
describe(bcgoff,"'+' cg ahead of ref. axis; data table is in scope read
format");
bchord = read("$FL_LOCAL_DB/tables/rotorcraft//blade_chord.tab");
describe(bchord,"Enter filename which is in scope read format");
bsegixx = read("$FL_LOCAL_DB/tables/rotorcraft//blade_rotary_mass.tab");
describe(bsegixx,"Enter filename which is in scope read format (m-l-l)");
btw = read("$FL_LOCAL_DB/tables/rotorcraft//blade_twist.tab");
describe(btw,"Enter filename which is in scope read format");
mpl = read("$FL_LOCAL_DB/tables/rotorcraft//mpl.tab");
describe(mpl,"Enter filename which is in scope read format (m/l)");
xnode = read("$FL_LOCAL_DB/tables/rotorcraft//blade_struc_xnode.tab");
describe(xnode,"If grip option is set to 0; Enter filename which is in
scope read format");
bsege0 = read("$FL_LOCAL_DB/tables/rotorcraft//blade_sege0.tab");
describe(bsege0,"Offset of midchord from elastic axis, +, e.a. ahead of
midchord");
//******* end loading data *******
popg // data

popg // rotor

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/blade/rigid/");
pushg (rotor)

pushg (data)
isRigid = 1;
popg // data

pushg (data)
//******* begin loading data *******
//******* end loading data *******
popg // data

popg // rotor

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/airloads/");
pushg (rotor)

pushg (data)
isAirloads = 1;
popg // data

pushg (data)
//******* begin loading data *******
nsega = 5 "Number of blade aero segments";
firstacpib = 3.502 "Inborad radial station where airfoil starts (l)";
//******* end loading data *******
popg // data
popg // rotor

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/airloads/quasi_steady/
");
pushg (rotor)

pushg (data)
isQuasiSteady = 1;
popg // data

pushg (data)
//******* begin loading data *******
nairfoil = 1 "number of airfoil used for each blade";
afoilboundary = read("$FL_LOCAL_DB/tables/rotorcraft//afoilboundary.tab");
describe(afoilboundary,"airfoil boundary, 1st col for inner edge; 2nd col
for outer edge; nd by radius");
load("$FL_LOCAL_DB/tables/rotorcraft//cll1.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//clh1.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//cml1.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//cmh1.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//cdl1.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//cdh1.sav");
aoatl1 = [-32  32  2  33];
aoath1 = [-180  180  2  181];
macht1 = [0  1.0  0.1  11];
load("$FL_LOCAL_DB/tables/rotorcraft//cll1.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//clh1.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//cml1.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//cmh1.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//cdl1.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//cdh1.sav");
aoatl2 = [-30  30  0.5  121];
aoath2 = [-180  180  5  73];
macht2 = [0  1.0  0.1  11];
load("$FL_LOCAL_DB/tables/rotorcraft//cll1.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//clh1.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//cml1.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//cmh1.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//cdl1.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//cdh1.sav");
aoatl3 = [-30  30  0.5  121];
aoath3 = [-180  180  5  73];
macht3 = [0  1.0  0.1  11];
//******* end loading data *******
popg // data

popg // rotor

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/induced_velocity/");
pushg (rotor)
pushg (data)
isInducedVelocity = 1;
popg // data

pushg (data)
//******* begin loading data *******
//******* end loading data *******
popg // data

popg // rotor

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/induced_velocity/unifo
rm/");
pushg (rotor)

pushg (data)
isUniformInflow = 1;
popg // data

pushg (data)
//******* begin loading data *******
gef1 = 0.5 "Empirical ground effect parameter";
gef2 = -.6667 "Empirical ground effect parameter";
dwtau = 0.0103 "Inflow time constant in seconds";
//******* end loading data *******
popg // data

popg // rotor
exec("applpath.scr",1); // application path
pushg (data)
      exec("rotor1.prolog",1);
popg // data
      exec("rotor1.exc",1);
popg // rotor1_rotor

path([])

exec("$FL_SCRIPTS/permpath.scr",1);

path("$FL_SCRIPTS/rotorcraft/rotors/");
pushg (rotor2_rotor)

pushg (data)
isRotor2 = 1;
popg // data

pushg (data)
//******* begin loading data *******
fsmr = 61.0 "Fuselage station of Rotor2 (1)";
blmr = -1.1667 "Buttline  station of Rotor2 (1)";
wlmr = 27.0583 "Waterline station of Rotor2 (1)";
```

```
nblades = 4 "Number of Rotor blades";
Btl = 0.92 "Rotor blade tip loss factor";
eulerhub = [0.0   0.0  -110.0];
tiltaxis = 1 "Axis about which rotor shaft tilts as in tilt rotor. 1:
x-axis; 2: y; 3: z";
rpmnom = 124.62 "Rotor nomial rotational speed in rad/sec";
rmr = 5.5 "Rotor radius in feet";
naz = 111 "Number of azimuth step/rev";
phase = 0.0 "swash plate phase angle (deg)";
dtopt = 0 "option 0: use the same dt as rotor1;  1: use different dt for
rotor2";
//******* end loading data *******
popg // data

path("$FL_SCRIPTS/rotorcraft/rotors/bailey_rotor/");
pushg (rotor)

pushg (data)
isBaileyRotor = 1;
popg // data

pushg (data)
//******* begin loading data *******
atr = 5.73 "Lift curve slope";
cdtr = 0.001 "Rotor head drag coefficient";
d0tr = 0.0087 "The zeroth order drag polar constant";
d1tr = -0.0216 "First order drag  polar constant";
d2tr = 0.4000 "Second order drag polar constant";
biastr = 6.0 "Blade pitch at zero collective";
bvttr = 0.796 "Blockage effect parameter for hover and low speed (less
than vbvttr)";
bvt1tr = .85 "Blockage effect parameter for high speed (larger than
vbvttr)";
chordtr = 0.81 "Blade chord";
delttr = 0.001455 "Partial of coning wrt thrust";
ttheta = 0.0 "Initial tail rotor collective pitch";
twsttr = -18 "Difference between blade tip and root pitch";
td3tr = 0.7002075 "Tan of effective delta 3 angle (for pitch to coning
coupling)";
vbvttr = 50.6340 "forward speed threshhold for blockage effect";
xibtr = 3.10 "Second moment of mass inertia";
drot = 1 "1: Rotational Axis(RA) = thrust Axis(TA); -1:  RA= -TA ";
//******* end loading data *******
popg // data

popg // rotor
exec("applpath.scr",1); // application path
pushg (data)
      exec("rotor2.prolog",1);
popg // data
      exec("rotor2.exc",1);
```

```
popg // rotor2_rotor

path([])

exec("$FL_SCRIPTS/permpath.scr",1);


path("$FL_SCRIPTS/rotorcraft/propulsion/");
pushg (propulsion)

pushg (data)
isPropulsion = 1;
popg // data

pushg (data)
//******* begin loading data *******
//******* end loading data *******
popg // data

path("$FL_SCRIPTS/rotorcraft/propulsion/ideal/");
pushg (propulsion)

pushg (data)
isIdeal = 1;
popg // data

pushg (data)
//******* begin loading data *******
psiscale = 4.6156 "Ratio of tail rotor rpm to main rotor rpm";
//******* end loading data *******
popg // data

popg // propulsion
exec("applpath.scr",1); // application path
pushg (data)
      exec("propulsion.prolog",1);
popg // data
      exec("propulsion.exc",1);
popg // propulsion

path([])

exec("$FL_SCRIPTS/permpath.scr",1);

path("$FL_SCRIPTS/rotorcraft/airframe/");
pushg (airframe)

pushg (data)
isAirframe = 1;
popg // data
```

```
pushg (data)
//******* begin loading data *******
//******* end loading data *******
popg // data

path("$FL_SCRIPTS/rotorcraft/airframe/fuselage/");
pushg (airframe)

pushg (data)
isFuselage = 1;
popg // data

pushg (data)
//******* begin loading data *******
//******* end loading data *******
popg // data

popg // airframe

path("$FL_SCRIPTS/rotorcraft/airframe/fuselage/rigid/");
pushg (airframe)

pushg (data)
isRigid = 1;
popg // data

pushg (data)
//******* begin loading data *******
//******* end loading data *******
popg // data

popg // airframe

path("$FL_SCRIPTS/rotorcraft/airframe/aerodynamics/");
pushg (airframe)

pushg (data)
isAero = 1;
popg // data

pushg (data)
//******* begin loading data *******
//******* end loading data *******
popg // data

popg // airframe

path("$FL_SCRIPTS/rotorcraft/airframe/aerodynamics/fuselage/");
pushg (airframe)

pushg (data)
```

```
isFuselageaero = 1;
popg // data

pushg (data)
//******* begin loading data *******
fswf = 28.7917 "Fuselage  station of fuselage body airloads acting point
(1) ";
blwf = 0.0 "Buttline  sttion of fuselage body airloads acting point (1)";
wlwf = 19.5 "Waterline station of elage body airloads acting point (1)";
faoat1 = [-30.0  30.0  5  13];
faoat2 = [-90.0  90.0  10  19];
faost1 = [-30.0  30.0  5  13];
faost2 = [-90.0  90.0  10  19];
faoat3 = [-10.0 10.0 10 3];
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fuseclal.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fuseclah.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fusecdal.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fusecdah.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fusecmal.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fusecmah.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fuseclbal.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fuseclbah.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fusecdbl.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fusecdbh.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fusecmbal.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fusecmbah.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fusecybal.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fusecybah.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fusecnbal.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fusecnbah.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fusecrbal.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60fusecrbah.sav");
//******* end loading data *******
popg // data

popg // airframe
path("$FL_SCRIPTS/rotorcraft/airframe/aerodynamics/htail/");
pushg (airframe)

pushg (data)
isHstabaero = 1;
popg // data

pushg (data)
//******* begin loading data *******
nhstab = 2 "Number of horizontal stabilizers";
fsht = [58.3416 58.3416];
blht = [3.5  -3.5];
wlht = [20.3667 20.3667];
hpsi = [90.0   90.0];
htheta = [0.0  0.0];
```

```
hphi = [180  180];
hstabidx = [1 -1];
hstabswp = [0.0   0.0];
hlen = [7.1667 7.1667];
//******* end loading data *******
popg // data

popg // airframe

path("$FL_SCRIPTS/rotorcraft/airframe/aerodynamics/htail/");
pushg (airframe)

pushg (data)
ishstabilator = 1;
popg // data

pushg (data)
//******* begin loading data *******
nhstabseg = [1.0 1.0];
hchord = read("$FL_LOCAL_DB/tables/rotorcraft//hstabchord.tab");
describe(hchord,"In scope read format. 1st two column for 1st hstab and so
on");
htailang = [0.0 0.0];
hdefic = [0.95 0.95];
haoatl = [-30 30 5 13];
haoath = [-90 90 10 19];
hmacht = [0  1  1  1];
load("$FL_LOCAL_DB/tables/rotorcraft//sh60htailcll.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60htailclh.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60htailcdl.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60htailcdh.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60htailcml.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60htailcmh.sav");
//******* end loading data *******
popg // data

popg // airframe

path("$FL_SCRIPTS/rotorcraft/airframe/aerodynamics/vtail/");
pushg (airframe)

pushg (data)
isVstabaero = 1;
popg // data

pushg (data)
//******* begin loading data *******
nvstab = 1 "Number of vertical stabilizers";
fsvt = [57.9167];
blvt = [0.0];
wlvt = [22.75];
```

```
vpsi = [0.0];
vtheta = [90.0];
vphi = [90.0];
vstabidx = [1];
vstabswp = [10];
vlen = [7.88];
//******* end loading data *******
popg // data

popg // airframe

path("$FL_SCRIPTS/rotorcraft/airframe/aerodynamics/vtail/");
pushg (airframe)

pushg (data)
isvstabilator = 1;
popg // data

pushg (data)
//******* begin loading data *******
nvstabseg = [1.0];
vchord = read("$FL_LOCAL_DB/tables/rotorcraft//vstabchord.tab");
describe(vchord,"In scope read format. 1st two column for 1st vstab and so
on");
vtailang = [0.0];
vdefic = [.95];
vaoatl = [-30  30  5  13];
vaoath = [-90 90 10  19];
vmacht = [0  1  1  1];
load("$FL_LOCAL_DB/tables/rotorcraft//sh60vtailcll.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60vtailclh.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60vtailcdl.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60vtailcdh.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60vtailcml.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//sh60vtailcmh.sav");
//******* end loading data *******
popg // data

popg // airframe

path("$FL_SCRIPTS/rotorcraft/airframe/sensor/");
pushg (airframe)

pushg (data)
isSensor = 1;
popg // data

pushg (data)
//******* begin loading data *******
fssens = 0.0 "Fuselage  station of sensor";
blsens = 0.0 "Buttline  station of sensor";
```

```
wlsens = 0.0 "Waterline station of sensor";
//******* end loading data *******
popg // data

popg // airframe

path("$FL_SCRIPTS/rotorcraft/airframe/landing_gears/");
pushg (airframe)

pushg (data)
isLandingGear = 1;
popg // data

pushg (data)
//******* begin loading data *******
fsrmg = 24.7858 "fuselage  station of right main gear ";
blrmg = 4.4333 "buttline  station of right main gear ";
wlrmg = 15.4000 "waterline  station of right main gear ";
fslmg = 24.7858 "fuselage  station of left main gear ";
bllmg = -4.4333 "buttline  station of left main gear ";
wllmg = 15.4000 "waterline  station of left main gear ";
fstg = 53.7183 "fuselage  station of tail gear ";
bltg = 0.0 "buttline  station of tail gear ";
wltg = 15.0333 "waterline  station of tail gear ";
brake = 3.0 "Brake pedal deflection ";
maxload = -40000.0 "Maximum gear reaction load";
mgmuside = -0.8 "Main gear side force  coefficient of friction";
mgudsl = 2.366 "Main gear undeflected strut length";
mgsl = 0.7 "Main gears' zero force deflection";
tgmuside = -0.8 "Tail gear side force  coefficient of friction";
tgudsl = 2.29 "Tail gear undeflected strut length";
tgsl = 0.94 "Tail gears' zero force deflection";
mgmub = read("$FL_LOCAL_DB/tables/rotorcraft//mgmub.tab");
describe(mgmub,"Braking coefficient of friction table (main gear)");
mgmur = read("$FL_LOCAL_DB/tables/rotorcraft//mgmur.tab");
describe(mgmur,"Rolling coefficient of friction table (main gear)");
tgmub = read("$FL_LOCAL_DB/tables/rotorcraft//tgmub.tab");
describe(tgmub,"Braking coefficient of friction table (tail gear)");
tgmur = read("$FL_LOCAL_DB/tables/rotorcraft//tgmur.tab");
describe(tgmur,"Rolling coefficient of friction table (tail gear)");
mgcc = read("$FL_LOCAL_DB/tables/rotorcraft//mgcc.tab");
describe(mgcc,"Two stage damping coefficient table(main gear)");
mgkk = read("$FL_LOCAL_DB/tables/rotorcraft//mgkk.tab");
describe(mgkk,"Two stage spring coefficient table(main gear)");
tgcc = read("$FL_LOCAL_DB/tables/rotorcraft//tgcc.tab");
describe(tgcc,"Two stage damping coefficient table(tail gear)");
tgkk = read("$FL_LOCAL_DB/tables/rotorcraft//tgkk.tab");
describe(tgkk,"Two stage spring coefficient table(tail gear)");
//******* end loading data *******
popg // data
popg // airframe
exec("applpath.scr",1); // application path
pushg (data)
      exec("airframe.prolog",1);
popg // data
      exec("airframe.exc",1);
popg // airframe

path([])

exec("$FL_SCRIPTS/permpath.scr",1);

path("$FL_SCRIPTS/rotorcraft/flight_control/");
pushg (control)

pushg (data)
isFlightControl = 1;
popg // data

pushg (data)
//******* begin loading data *******
//******* end loading data *******
popg // data

path("$FL_SCRIPTS/rotorcraft/flight_control/uh60/");
pushg (control)

pushg (data)
isUH60Control = 1;
popg // data

pushg (data)
//******* begin loading data *******
Xatrm = 5.16 "Preset Lateral Stick Postion";
Xbtrm = 4.73 "Preset Long Stick Postion";
Xctrm = 4.14 "Preset Coll Stick Postion";
Xptrm = 1.65 "Preset Pedal Stick Postion";
sensorno = 7 "Airframe sensor number";
sensorgain = read("$FL_LOCAL_DB/tables/rotorcraft//sensorgain.tab");
describe(sensorgain,"Airframe sensor selective matrix");
//******* end loading data *******
popg // data

popg // control
exec("applpath.scr",1); // application path
pushg (data)
      exec("flight_control.prolog",1);
popg // data
      exec("flight_control.exc",1);
popg // control
```

```
path([])

exec("$FL_SCRIPTS/permpath.scr",1);

path("$FL_SCRIPTS/rotorcraft/aero_interference/");
pushg (interference)

pushg (data)
isAeroInterference = 1;
popg // data

pushg (data)
//******* begin loading data *******
intfmatrix = read("$FL_LOCAL_DB/tables/rotorcraft//intfmatrix.tab");
describe(intfmatrix,"interference coupling matrix");
//******* end loading data *******
popg // data

exec("applpath.scr",1); // application path

pushg (data)
      exec("aero_interference.prolog",1);
popg // data
      exec("aero_interference.exc",1);

path([])

exec("$FL_SCRIPTS/permpath.scr",1);

path("$FL_SCRIPTS/rotorcraft/aero_interference/rotor/");
pushg (rotor1_rotor)

pushg (data)
isIntfRotor1 = 1;
popg // data

pushg (data)
//******* begin loading data *******
//******* end loading data *******
popg // data

path("$FL_SCRIPTS/rotorcraft/aero_interference/rotor/empirical/");
pushg (rotor1_rotor)

pushg (data)
isRotor1Empirical = 1;
popg // data

pushg (data)
//******* begin loading data *******
//******* end loading data *******
```

```
popg // data

popg // rotor1_rotor

path("$FL_SCRIPTS/rotorcraft/aero_interference/rotor/empirical/");
pushg (rotor1_rotor)
pushg (data)
isRotor1intfHtail = 1;
popg // data

pushg (data)
//******* begin loading data *******
htchit = [0  100  10  11];
hta1ft = [-6  6  6  3];
load("$FL_LOCAL_DB/tables/rotorcraft//mrint_htailx.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//mrint_htaily.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//mrint_htailz.sav");
//******* end loading data *******
popg // data

popg // rotor1_rotor

path("$FL_SCRIPTS/rotorcraft/aero_interference/rotor/empirical/");
pushg (rotor1_rotor)

pushg (data)
isRotor1intfVtail = 1;
popg // data

pushg (data)
//******* begin loading data *******
vtchit = [0  100  10  11];
vta1ft = [-6  6  6  3];
load("$FL_LOCAL_DB/tables/rotorcraft//rotor1intf_vtailvx.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//rotor1intf_vtailvy.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//rotor1intf_vtailvz.sav");
//******* end loading data *******
popg // data

popg // rotor1_rotor

path("$FL_SCRIPTS/rotorcraft/aero_interference/rotor/empirical/");
pushg (rotor1_rotor)

pushg (data)
isRotor1intfFuse = 1;
popg // data

pushg (data)
//******* begin loading data *******
fusechit = [0  100  10  11];
```

```
fusealft = [-6  6  6  3];
load("$FL_LOCAL_DB/tables/rotorcraft//mrint_fusex.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//mrint_fusey.sav");
load("$FL_LOCAL_DB/tables/rotorcraft//mrint_fusez.sav");
//******* end loading data *******
popg // data

popg // rotor1_rotor
exec("applpath.scr",1); // application path
pushg (data)
      exec("intfrotor1.prolog",1);
popg // data
      exec("intfrotor1.exc",1);
popg // rotor1_rotor

path([])

exec("$FL_SCRIPTS/permpath.scr",1);

path("$FL_SCRIPTS/rotorcraft/aero_interference/fuselage/");
pushg (fuselage)

pushg (data)
isIntfFuselage = 1;
popg // data

pushg (data)
//******* begin loading data *******
//******* end loading data *******
popg // data

path("$FL_SCRIPTS/rotorcraft/aero_interference/fuselage/empirical/");
pushg (fuselage)

pushg (data)
isFuseEmpirical = 1;
popg // data

pushg (data)
//******* begin loading data *******
//******* end loading data *******
popg // data

popg // fuselage

path("$FL_SCRIPTS/rotorcraft/aero_interference/fuselage/empirical/");
pushg (fuselage)

pushg (data)
isFuseintfHtail = 1;
popg // data
```

```
pushg (data)
//******* begin loading data *******
htaost1 = [0 0 0 1];
htaoat1 = [-30  30  5  13];
load("$FL_LOCAL_DB/tables/rotorcraft//fuseintf_stabqdyn.sav");
htaost2 = [0 0 0 1];
htaoat2 = [-30 30 5 13];
load("$FL_LOCAL_DB/tables/rotorcraft//fuseintf_stabvz.sav");
htaost3 = [-90  90  180  2];
htaoat3 = [-90  90  180  2];
load("$FL_LOCAL_DB/tables/rotorcraft//fuseintf_htailvy.sav");
//******* end loading data *******
popg // data

popg // fuselage

path("$FL_SCRIPTS/rotorcraft/aero_interference/fuselage/empirical/");
pushg (fuselage)

pushg (data)
isFuseintfVtail = 1;
popg // data

pushg (data)
//******* begin loading data *******
vtaost1 = [-35  35 5  15];
vtaoat1 = [-10 10 10 3];
load("$FL_LOCAL_DB/tables/rotorcraft//fuseintf_vertqdyn.sav");
vtaost2 = [-90  90   180  2];
vtaoat2 = [-90  90  180  2];
load("$FL_LOCAL_DB/tables/rotorcraft//fuseintf_vtailvz.sav");
vtaost3 = [-30 30 5 13];
vtaoat3 = [-10 10 10 3];
load("$FL_LOCAL_DB/tables/rotorcraft//fuseintf_vertvy.sav");
//******* end loading data *******
popg // data

popg // fuselage
exec("applpath.scr",1); // application path
pushg (data)
      exec("intffuse.prolog",1);
popg // data
      exec("intffuse.exc",1);
popg // fuselage

path([])

exec("$FL_SCRIPTS/permpath.scr",1);

popg //interference
popg //model

// Second pass instructions: making connections

goto world
init;

path("$FL_SCRIPTS/rotorcraft/");
pushg (model)

path("$FL_SCRIPTS/rotorcraft/environ_atmos/");
pushg (model)

path("$FL_SCRIPTS/rotorcraft/environ_atmos/");
exec("applpath.scr",1); // application path
      exec("environ_atmos.epilog",1);
pushg (cpg);
      exec("environ_atmos.configure",1);
popg; // cpg
popg; // model

path([])

exec("$FL_SCRIPTS/permpath.scr",1);

path("$FL_SCRIPTS/rotorcraft/rotors/");
pushg (rotor1_rotor)

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/");

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/hub/");

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/hub/articulated/");

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/hub/articulated/lagdam
per/linear/");

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/blade/");

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/blade/rigid/");

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/airloads/");

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/airloads/quasi_steady/
");

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/induced_velocity/");

path("$FL_SCRIPTS/rotorcraft/rotors/blade_element/induced_velocity/unifo
rm/");
exec("applpath.scr",1); // application path
```

```
      exec("rotor1.epilog",1);
pushg (cpg);
      exec("rotor1.configure",1);
popg; // cpg
popg; // rotor1_rotor

path([])

exec("$FL_SCRIPTS/permpath.scr",1);


path("$FL_SCRIPTS/rotorcraft/rotors/");
pushg (rotor2_rotor)

path("$FL_SCRIPTS/rotorcraft/rotors/bailey_rotor/");
exec("applpath.scr",1); // application path
      exec("rotor2.epilog",1);
pushg (cpg);
      exec("rotor2.configure",1);
popg; // cpg
popg; // rotor2_rotor

path([])

exec("$FL_SCRIPTS/permpath.scr",1);

path("$FL_SCRIPTS/rotorcraft/propulsion/");
pushg (propulsion)

path("$FL_SCRIPTS/rotorcraft/propulsion/ideal/");
exec("applpath.scr",1); // application path
      exec("propulsion.epilog",1);
pushg (cpg);
      exec("propulsion.configure",1);
popg; // cpg
popg; // propulsion

path([])

exec("$FL_SCRIPTS/permpath.scr",1);

path("$FL_SCRIPTS/rotorcraft/airframe/");
pushg (airframe)

path("$FL_SCRIPTS/rotorcraft/airframe/fuselage/");

path("$FL_SCRIPTS/rotorcraft/airframe/fuselage/rigid/");

path("$FL_SCRIPTS/rotorcraft/airframe/aerodynamics/");

path("$FL_SCRIPTS/rotorcraft/airframe/aerodynamics/fuselage/");
path("$FL_SCRIPTS/rotorcraft/airframe/aerodynamics/htail/");

path("$FL_SCRIPTS/rotorcraft/airframe/aerodynamics/htail/");

path("$FL_SCRIPTS/rotorcraft/airframe/aerodynamics/vtail/");

path("$FL_SCRIPTS/rotorcraft/airframe/aerodynamics/vtail/");

path("$FL_SCRIPTS/rotorcraft/airframe/sensor/");

path("$FL_SCRIPTS/rotorcraft/airframe/landing_gears/");
exec("applpath.scr",1); // application path
      exec("airframe.epilog",1);
pushg (cpg);
      exec("airframe.configure",1);
popg; // cpg
popg; // airframe

path([])

exec("$FL_SCRIPTS/permpath.scr",1);

path("$FL_SCRIPTS/rotorcraft/flight_control/");
pushg (control)

path("$FL_SCRIPTS/rotorcraft/flight_control/uh60/");
exec("applpath.scr",1); // application path
      exec("flight_control.epilog",1);
pushg (cpg);
      exec("flight_control.configure",1);
popg; // cpg
popg; // control

path([])

exec("$FL_SCRIPTS/permpath.scr",1);


path("$FL_SCRIPTS/rotorcraft/aero_interference/");
pushg (interference)

path("$FL_SCRIPTS/rotorcraft/aero_interference/rotor/");
pushg (rotor1_rotor)

path("$FL_SCRIPTS/rotorcraft/aero_interference/rotor/empirical/");

path("$FL_SCRIPTS/rotorcraft/aero_interference/rotor/empirical/");

path("$FL_SCRIPTS/rotorcraft/aero_interference/rotor/empirical/");

path("$FL_SCRIPTS/rotorcraft/aero_interference/rotor/empirical/");
```

```
exec("applpath.scr",1); // application path
      exec("intfrotor1.epilog",1);
pushg (cpg);
      exec("intfrotor1.configure",1);
popg; // cpg
popg; // rotor1_rotor

path([])

exec("$FL_SCRIPTS/permpath.scr",1);

path("$FL_SCRIPTS/rotorcraft/aero_interference/fuselage/");
pushg (fuselage)

path("$FL_SCRIPTS/rotorcraft/aero_interference/fuselage/empirical/");

path("$FL_SCRIPTS/rotorcraft/aero_interference/fuselage/empirical/");

path("$FL_SCRIPTS/rotorcraft/aero_interference/fuselage/empirical/");
exec("applpath.scr",1); // application path
      exec("intffuse.epilog",1);
pushg (cpg);
      exec("intffuse.configure",1);
popg; // cpg
popg; // fuselage

path([])

exec("$FL_SCRIPTS/permpath.scr",1);

exec("applpath.scr",1); // application path
      exec("aero_interference.epilog",1);
pushg (cpg);
      exec("aero_interference.configure",1);
popg; // cpg
popg; // interference

path([])

exec("$FL_SCRIPTS/permpath.scr",1);

exec("applpath.scr",1); // application path
      exec("rotorcraft.epilog",1);
pushg (cpg);
      exec("rotorcraft.configure",1);
popg; // cpg
popg; // model

path([])

exec("$FL_SCRIPTS/permpath.scr",1);
exec("applpath.scr",1); // application path
exec("slv.scr",1);

goto world;      // world
```

The following is the data entered into Model Editor for the construction of the SH-60B model. It follows a one to one correspondence with the model tree.

Rotorcraft	
test flag	0 (total aircraft)
mass flag	0 (fuselage only)
vehicle mass	578.94 slugs
fuselage station of vehicle cg	29.508 ft
buttline station of vehicle cg	.0167 ft
waterline station of vehicle cg	20.7583 ft
roll moment of vehicel inertia Ixx	6135 slug-ft-ft
pitch moment of vehicle inertia Iyy	48715 slug-ft-ft
yaw moment of vehicle inertia Izz	46785 slug-ft-ft
X-Y axis product of vehicle inertia	-317
X-Z axis product of vehicle inertia	2535
Y-Z axis product of vehicle inertia	-195

EnvironAtmos
No data needed

Rotorcraft.EnvironAtmos.Atmosphere	
atmosphere table	atmo.tab

Rotorcraft.Rotor1	
fuselage station of rotor1	28.4346 feet
buttline station of rotor1	0.0 ft
waterline station of rotor1	26.25 ft
number of rotor1 blades	4
rotor blade tip loss factor	.97
rotor hub frame orientation in Euler	0.0 177.0 0.0
axis about which shaft tilts	2 (y axis)
rotor1 nominal speed	27.0 rad/sec
rotor1 radius	26.83 ft
number of azimuth step/rev	24
swash plate phase angle	-9.7 deg

Rotorcraft.Rotor1.Blade Element	
flag for counterclockwise	0 CCW

Rotorcraft.Rotor1.Blade Element.Hub	
rotor precone	0.0 deg
precone offset from rotation center	0.0 ft
torque offset	0.0 ft

Rotorcraft.Rotor1.Blade Element.Hub.Articulated	
lag hinge option	1 (with lag hinge)
flap hinge offset	1.25 ft
lag hinge offset	1.25 ft

```
feathering hinge offset               1.25 ft
delta 3                               0.0 deg
spring stiffness of flap hinge              0.0 ft-ibf/rad
damping coefficient of flap hinge     0.0 ft-ibf.sec/rad
flap hinge spring undeformed angle    0.0 deg

Rotor1.Blade Element.Hub.Articulated.Lineardamper
linear lag damper spring stiffness    0 ft-ibf/rad
linear lag damper damping coefficient       20000 ft-ibf.sec/rad
linear lag damper spring udeformed    0 deg


Rotorcraft.Rotor1.Blade Element.Blade
inboard edge location of blade struc        1.25 ft
blade structure grid generation option      1
blade section chordwise c.g. offset blade_cgoff.tab
blade chord                     blade_chord.tab
blade segment rotary mass             blade_rotary_mass.tab
blade twist variation                 blade_twist.tab
blade mass distribution               mpl.tab
blade structural segment end nodes    blade_struc_xnode.tab
offset of midchord from e.a           blade_sege0.tab
   offset of midchord from elastic axis,+, e.a. ahead of midchord

Rotorcraft.Rotor1.Blade Element.Rigid
No data needed

Rotorcraft.Rotor1.Blade Element.Airloads
number of blade aero segments         5
inboard radial station where airfoil starts      3.502 ft

Rotorcraft.Rotor1.Blade Element.Airloads.QuasiSteady
number of airfoils                    1
airfoil boundary                afoilboundary.tab
1st airfoil cl for low angle          cll1.sav
1st airfoil cl for high angle         clh1.sav
1st airfoil cm for low angle          cml1.sav
1st airfoil cm for high angle         cmh1.sav
1st airfoil cd for low angle          cdl1.sav
1st airfoil cd for high angle         cdh1.sav
1st airfoil low angle tab arguments -32   32  2   33
1st airfoil high angle tab arguments       -180   180   2 181
1st airfoil mach number arguments   0  1.0  0.1  11

Rotorcraft.Rotor1.Blade Element.InducedVelocity
No data needed

Rotorcraft.Rotor1.Blade Element.InducedVelocity.UniformInflow
empirical ground effect parameter   .5
empirical ground effect parameter   -.6667
inflow time constant                .0103
```

```
Rotorcraft.Rotor2
fuselage station of rotor2           61.0 ft
buttline station of rotor2           -1.1667 ft
waterline station of rotor2          27.0583 ft
number of rotor blades               4
rotor blade tip loss factor          .92
rotor hub frame orientation in euler       0.0 0.0 -110.0
axis about which the shaft tilts     1 y axis
rotor nominal speed                  124.62 rad/sec
rotor radius                         5.5 ft
number of azimuth step/rev           111
swash plate phase angle              0.0
option for integration time step     0 use same dt as rotor1

Rotorcraft.Rotor2.BaileyRotor
lift curve slope               5.73
rotor head drag coefficient          2.20
constant coeff of drag               .0087
first order drag polar constant            -.0216
second order drag polar constant     .4
blade pitch bias               6.0
blockage effect for low speed        .796
blockage effect for high speed             .85
blade chord                     .81 ft
partial of coning wrt thrust         .001455 deg/lbf
tail rotor collective pitch          0.0 rad
tail rotor blade twist               -18.0 deg
tan of delta 3 angle                 .7002075 (tan 35 deg)
forward speed thrshold for blockage 50.6340
second moment of mass inertia        3.10
direction of rotation axis           1 rotational axis(RA)=
                                thrust axis(TA)

Rotorcraft.Propulsion
No data needed

Rotorcraft.Propulsion.Ideal
RPM ratio of tail rotor to main rotor     4.6156

Rotorcraft.Airframe
No data needed

Rotorcraft.Airframe.Fuselage
No data needed

Rotorcraft.Airframe.Fuselage.Rigid
No data needed

Rotorcraft.Airframe.Aero
No data needed
```

```
Rotorcraft.Airframe.Aero.Fuselagaero
fuselage station of fuselage aero   28.75 ft
buttline station of fuselage aero   0.0 ft
waterline station of fuselage aero  19.5 ft
alpha arguments of low alpha table  -30.0  30.0  5  13
alpha arguments of high alpha table -90.0  90.0  10  19
beta arguments of low beta table    -30.0  30.0  5  13
beta arguments of high beta table   -90.0  90.0  10  19
cross coupling a.o.a. table arguments     -10.0 10.0 10 3
low alpha table of lift             sh60fuseclal.sav
high alpha table of lift            sh60fuseclah.sav
low alpha table of drag             sh60fusecdal.sav
high alpha table of drag            sh60fusecdah.sav
low alpha table of pitch moment           sh60fusecmal.sav
high alpha table of pitch moment    sh60fusecmah.sav
low beta table of lift from side slip     sh60fuseclbal.sav
                  coupling
high beta table of lift from side slip    sh60fuseclbah.sav
                  coupling
low beta table of drag              sh60fusecdbl.sav
high beta table of drag             sh60fusecdbh.sav
low beta table of pitch moment            sh60fusecmbal.sav
high beta table of pitch moment           sh60fusecmbah.sav
low beta table of side force        sh60fusecybal.sav
high beta table of side force       sh60fusecybah.sav
low beta table of roll moment       sh60fusecnbal.sav
high beta table of roll moment            sh60fusecnbah.sav
low beta table of yaw moment        sh60fusecrbal.sav
high beta table of yaw moment       sh60fusecrbah.sav

Rotorcraft.Airframe.Aero.Hstabaero
Number of hstab                     2
fuselage station of h-tail          58.3416 58.3416ft
buttline station of h-tail          3.5 -3.5 ft
waterline station of h-tail         20.3667  20.3667 ft
hstab local frame orientation Psi   90.0 90.0 degs
hstab local frame orientation Theta 0.0 0.0 degs
hstab local frame orientation Phi   180 180 degs
ID for local x-axis direction       1 -1
hstab sweep angle             0.0 0.0
hstab length                        7.1667 7.1667 ft

Rotorcraft.Airframe.Aero.Hstabaero.hstabilator
number of airload segments          1.0 1.0
chord of hstab segments             hstabchord.tab
initial incidence of angel of h-tail      0.0 0.0 deg
lift deficency factor               .95 .95
alpha arguments of low angel table  -30 30 5 13 deg
alpha arguments of high angle table -90 90 10 19 deg
mach number arguments of low angle table  0 1 1 1
low angle table of cl               sh60htailcll.sav
```

```
high angle table of cl              sh60htailclh.sav
low angle table of cd               sh60htailcdl.sav
high angle table of cd              sh60htailcdh.sav
low angle table of cm               sh60htailcml.sav
high angle table of cm              sh60htailcmh.sav

Rotorcraft.Airframe.Aero.Vstabaero
number of vstab                     1
vstab fuselage station              57.9167 ft
vstab buttline station              0.0 ft
vstab waterline station             22.75
vstab local frame orientation Psi   0.0
vstab local frame orientation Theta 90.0 degs
vstab local frame orientation Phi   90.0 degs
ID for local x-axis direction       1
vstab sweep angle             10 degs
vstab length                        7.88 ft

Rotorcraft.Airframe.Aero.Vstabaero.vstabilator
number of airload segments          1.0
vstab chord variation               vstabchord.tab
incidence angle of vstab            0.0 degs
lift deficiency factor              .95
alpha arguments of low angle table  -30 30 5 13
alpha arguments of high angle table -90 -90 10 19
mach number arguments               0 1 1 1
low angle table of cl               sh60vtailcll.sav
high angle table of cl              sh60vtailclh.sav
low angel table of cd               sh60vtailcdl.sav
high angle table of cd              sh60vtailcdh.sav
low angle table of cm               sh60vtailcml.sav
high angle table of cm              sh60vtailcmh.sav

Rotorcraft.Airframe.Sensor
fuselage station of sensor          0.0 ft
buttline station of sensor          0.0 ft
waterline station of sensor         0.0 ft

Rotorcraft.Airframe.Landing Gear
fuselage station of right gear           24.7858
buttline station of right gear           4.4333
waterline station of right gear          15.4000
fuselage station of left gear      24.7858
buttline station of left gear      -4.4333
waterline sation of left gear      15.4000
fuselage sation of tail gear       53.7183
buttline station of tail gear      0.0
waterline staion of tail gear      15.0333
brake pedal deflection             3.0 ft
maximum gear reaction load         -40000.0 ibf
main gear side friction force coeff -0.8 ibf.sec/ft
main gears undeflected strut length 2.366 ft
main gears zero force deflection    .7
tail gear side friction force coeff -.8
tail gears undeflected strut length 2.29 ft
tail gears zero force deflection    .94 ft
braking friction coeff table MG           mgmub.tab
rolling friction coeff table MG           mgmur.tab
braking friction coeff table TG           tgmub.tab
rolling friction coeff table TG           tgmur.tab
damping coeff table MG              mgcc.tab
spring coeff table MG               mgkk.tab
damping coeff table TG              tgcc.tab
spring coeff table TG               tgkk.tab

Rotorcraft.Flight Control.UH60 Control
lat stick                     5.16
long stick                    4.73 in
coll. stick                   4.14 in
pedal                         1.65
sensor number                       7
sensor selective matrix             sensorgain.tab

Rotorcraft.Aero.Interference
interference matrix                 intfmatrix.tab

Rotorcraft.Aero Interference.Intf Rotor1
No data needed

Rotorcraft.Aero Interference.Intf Rotor1.Rotor1 Empirical.Rotor1intfHtail
rotor dwash tab arg(wake skew angle)      0 100 10 11 degs
rotor dwash tab arg(tpp lng tilt ang)     -6 6 6 3 deg
rotor dwash x-comp table            mrint_htailx.sav
rotor dwash y-comp table            mrint_htaily.sav
rotor dwash z-comp table            mrint_htailz.sav

Rotorcraft.Aero Interference.Intf Rotor1.Rotor1 Empirical.Rotor1intfVtail
rotor dwash tab arg(wake skew angle)      0 100 10 11 degs
rotor dwash tab arg(tpp lng tilt ang)     -6 6 6 3 deg
rotor dwash x-comp table            mrintf_vtailx.sav
rotor dwash y-comp table            mrintf_vtaily.sav
rotor dwash z-comp table            mrintf_vtailz.sav

Rotorcraft.Aero Interference.Intf Rotor1.Rotor1 Empirical.Rotor1intfFuse
rotor dwash tab arg(wake skew angle)      0 100 10 11 degs
rotor dwash tab arg(tpp lng tilt ang)     -6 6 6 3 deg
rotor dwash x-comp table            mrintf_fusex.sav
rotor dwash y-comp table            mrintf_fusey.sav
rotor dwash z-comp table            mrintf_fusez.sav

Rotorcraft.Aero Interference.Intf Fuselage.FuseEmpirical.FuseintfHtail
dy pressure loss tab arg(beta)            0 0 0 1 deg
dy pressure loss tab arg(alpha)             -30 30 5 13 deg
dy pressure loss table                 fuseintf_stabqdyn.sav
fuselage dwash tab arg(beta)           0 0 0 1 deg
fuselage dwash tab arg(alpha)          -30 30 5 13 deg
fuselage dwash table                   fuseintf_stabvz.sav
fuselage sidewash tab arg(beta)             -90  90  180  2 deg
fuselage sidewash tab arg(alpha)       -90  90  180  2
fuselage sidewash table                fuseintf_htailvy.sav

Rotorcraft.Aero Interference.Intf Fuselage.FuseEmpirical.FuseintfVtail
dy pressure loss tab arg(beta)              -35  35 5  15 deg
dy pressure loss tab arg(alpha)             -10 10 10 3 deg
dy pressure loss table                 fuseintf_vertqdyn.sav
fuselage dwash tab arg(beta)           -90  90   180  2 deg
fuselage dwash tab arg(alpha)          -90  90  180  2 deg
fuselage dwash table                   fuseintf_vtailvz.sav
fuselage sidewash tab arg(beta)             -30 30 5 13 deg
fuselage sidewash tab arg(alpha)       -10 10 10 3 deg
fuselage sidewash table                fuseintf_vertvy.sav

////////////////////////////////////////////////////
//
// file applpath.scr
//
// This file sets a path so that control system script files
//    control.exc, contol.prolog, control.epilog, and
//    control.configure can be found by sh60b_rigid.def
//
////////////////////////////////////////////////////
path("~/sh60")
// end of applpath.scr
```

```
/////////////////////////////////////////////////////
// file control.exc
//   this file is created in model editor by the model
//   sh60control.mod
/////////////////////////////////////////////////////
GOTO WORLD
GROUP MODEL
GROUP CONTROL
GROUP Sensors
BLIMITER VDOTLIMIT;
BLIMITER RLIMIT;
BLIMITER QLIMIT;
BLIMITER PLIMIT;
BSOURCE DOFSENSORA;
BNGAIN psi;
BNGAIN theta;
BNGAIN phi;
BNGAIN r;
BNGAIN q;
BNGAIN p;
BGAIN psideg;
BGAIN thetadeg;
BGAIN phideg;
BGAIN rdeg;
BGAIN qdeg;
BGAIN pdeg;
BSOURCE ALPHASRC;
BSOURCE BETASRC;
BSOURCE UNITY;
BSOURCE DOFSENSORL;
BNGAIN U;
BNGAIN V;
BNGAIN W;
BNGAIN VDOT;
BGAIN UKNOTS;
BGAIN VKNOTS;
BGAIN WKNOTS;
BTRANSF PSENSOR;
BTRANSF QSENSOR;
BTRANSF RSENSOR;
BSINK ALPHA;
BSINK BETA;
BTRANSF ALPHADELAY;
BTRANSF BETADELAY;
BTRANSF LATACCEL;
CONNECT RLIMIT(1) TO RSENSOR(1);
CONNECT VDOTLIMIT(1) TO LATACCEL(1);
CONNECT VDOT(1) TO VDOTLIMIT(1);
CONNECT DOFSENSORL(1) TO VDOT(1);
CONNECT QLIMIT(1) TO QSENSOR(1);
CONNECT qdeg(1) TO QLIMIT(1);
CONNECT rdeg(1) TO RLIMIT(1);
CONNECT PLIMIT(1) TO PSENSOR(1);
CONNECT pdeg(1) TO PLIMIT(1);
CONNECT r(1) TO rdeg(1);
CONNECT q(1) TO qdeg(1);
CONNECT p(1) TO pdeg(1);
CONNECT V(1) TO VKNOTS(1);
CONNECT W(1) TO WKNOTS(1);
CONNECT DOFSENSORA(1) TO psi(1);
CONNECT DOFSENSORA(1) TO theta(1);
CONNECT DOFSENSORA(1) TO phi(1);
CONNECT DOFSENSORA(1) TO r(1);
CONNECT DOFSENSORA(1) TO q(1);
CONNECT DOFSENSORA(1) TO p(1);
CONNECT psi(1) TO psideg(1);
CONNECT theta(1) TO thetadeg(1);
CONNECT phi(1) TO phideg(1);
CONNECT DOFSENSORL(1) TO U(1);
CONNECT DOFSENSORL(1) TO V(1);
CONNECT DOFSENSORL(1) TO W(1);
CONNECT U(1) TO UKNOTS(1);
CONNECT BETASRC(1) TO BETADELAY(1);
CONNECT BETADELAY(1) TO BETA(1);
CONNECT ALPHASRC(1) TO ALPHADELAY(1);
CONNECT ALPHADELAY(1) TO ALPHA(1);


PARENTG
GROUP Stabilator
BGAIN KTS2FPS2;
BNGAIN DEG2RAD;
BSINK STABINPUTSINK;
BNSUMJ STABINPUT;
BTRANSF VXLAG;
BVGAIN STABPOSITMAP;
BSOURCE UNITY;
BVGAIN GAINMAP;
BPRODJ PROD1;
BSUMJ SUMJ1;
BSINK STABPOSIT;
BGAIN GAIN1;
BSUMJ SUMJ2;
BLIMITER RANGELIMIT;
BTRANSF ACTUATOR;
BSOURCE VECTORUNITY;
BGAIN XCPC;
CONNECT KTS2FPS2(1) TO PROD1(1);
CONNECT DEG2RAD(1) TO STABPOSIT(1);
CONNECT RANGELIMIT(1) TO DEG2RAD(1);
CONNECT STABINPUT(1) TO STABINPUTSINK(1);
CONNECT VXLAG(1) TO STABINPUT(1);
CONNECT STABPOSITMAP(1) TO SUMJ1(2);
CONNECT GAINMAP(1) TO PROD1(2);
CONNECT PROD1(1) TO SUMJ1(1);
CONNECT SUMJ1(1) TO SUMJ2(1);
CONNECT SUMJ2(1) TO ACTUATOR(1);
CONNECT GAIN1(1) TO SUMJ2(2);
CONNECT ACTUATOR(1) TO RANGELIMIT(1);
CONNECT UNITY(1) TO GAINMAP(1);
CONNECT VECTORUNITY(1) TO STABPOSITMAP(1);
CONNECT XCPC(1) TO STABINPUT(2);


PARENTG
GROUP FlightControls
GROUP Collective
BLIMITER THETLIMIT;
BSOURCE THOCHK;
BSUMJ SUMJ4;
BSINK THETA0OUT;
BGAIN BIAS;
BSUMJ SUMJ3;
BGAIN KXCTH0;
BSOURCE XCILS  ;
BSUMJ SUMJ2;
BSUMJ SUMJ1;
BSOURCE XC;
BSOURCE XCTRM;
BGAIN DEG2RAD;
BTRANSF PRISERVO;
CONNECT THOCHK(1) TO SUMJ4(2);
CONNECT SUMJ4(1) TO DEG2RAD(1);
CONNECT XC(1) TO SUMJ1(1);
CONNECT XCTRM(1) TO SUMJ1(2);
CONNECT SUMJ1(1) TO THETLIMIT(1);
CONNECT THETLIMIT(1) TO SUMJ2(1);
CONNECT XCILS  (1) TO SUMJ2(2);
CONNECT SUMJ2(1) TO KXCTH0(1);
CONNECT KXCTH0(1) TO SUMJ3(1);
CONNECT BIAS(1) TO SUMJ3(2);
CONNECT DEG2RAD(1) TO THETA0OUT(1);
CONNECT SUMJ3(1) TO PRISERVO(1);
CONNECT PRISERVO(1) TO SUMJ4(1);


PARENTG
GROUP Longitudinal
BSUMJ SUMJ4;
BSUMJ SUMJ8;
BSOURCE B1SCHK;
BSUMJ SUMJ6;
BSOURCE XB;
```

```
BSUMJ SUMJ1;
BLIMITER B1SLIMIT;
BSUMJ SUMJ3;
BGAIN KXBB1S;
BSUMJ SUMJ5;
BGAIN BIAS;
BSINK B1SOUT;
BGAIN DEG2RAD;
BSUMJ SUMJ7;
GROUP PitchAnalogSAS
BGAIN PAGROUND;
BDISW DSAS;
BDISW ASAS;
BSUMJ PASUM2;
BGAIN PAGAIN3;
BLIMITER PALIMIT;
BTRANSF PALAG;
BSUMJ PASUM1;
BGAIN PAGAIN1;
BGAIN PAGAIN2;
BTRANSF PAWASHOUT;
CONNECT PASUM2(1) TO PALIMIT(1);
CONNECT ASAS(1) TO PAGAIN3(1);
CONNECT PAGAIN3(1) TO PASUM2(1);
CONNECT PAGROUND(1) TO DSAS(1);
CONNECT PAGROUND(1) TO ASAS(1);
CONNECT ASAS(1) TO DSAS(2);
CONNECT DSAS(1) TO PASUM2(2);
CONNECT PALAG(1) TO PASUM1(2);
CONNECT PASUM1(1) TO ASAS(2);
CONNECT PAGAIN1(1) TO PALAG(1);
CONNECT PAGAIN2(1) TO PASUM1(1);
CONNECT PAWASHOUT(1) TO PAGAIN1(1);
CONNECT PAWASHOUT(1) TO PAGAIN2(1);


PARENTG
GROUP PitchDigitalSAS
BGAIN PDGROUND;
BDISW ASAS;
BDISW DSAS;
BSUMJ PDSUM2;
BGAIN PDGAIN3;
BLIMITER PDLIMIT;
BSUMJ PDSUM1;
BTRANSF PDLAG;
BGAIN PDGAIN1;
BGAIN PDGAIN2;
BTRANSF PDWASHOUT;
CONNECT PDGAIN1(1) TO PDLAG(1);
CONNECT PDWASHOUT(1) TO PDGAIN1(1);
CONNECT PDSUM2(1) TO PDLIMIT(1);
CONNECT PDGROUND(1) TO ASAS(1);
CONNECT PDGROUND(1) TO DSAS(1);
CONNECT DSAS(1) TO ASAS(2);
CONNECT ASAS(1) TO PDSUM2(2);
CONNECT DSAS(1) TO PDGAIN3(1);
CONNECT PDGAIN3(1) TO PDSUM2(1);
CONNECT PDSUM1(1) TO DSAS(2);
CONNECT PDGAIN2(1) TO PDSUM1(2);
CONNECT PDLAG(1) TO PDSUM1(1);
CONNECT PDWASHOUT(1) TO PDGAIN2(1);


PARENTG
GROUP PitchSASInput
BGAIN DEG2IN;
BGAIN PER2DEG;
BLIMITER PSASLIMIT;
BTRANSF PSASACTUATOR;
CONNECT PSASLIMIT(1) TO PER2DEG(1);
CONNECT PER2DEG(1) TO DEG2IN(1);
CONNECT PSASACTUATOR(1) TO PSASLIMIT(1);


PARENTG
BSUMJ SUMJ2;
GROUP PBA
BSOURCE ZERO;
BDISW PBASWITCH;
BGAIN PBABIAS;
BLIMITER PBATHETLIMIT;
BLIMITER PBAQLIMIT;
BGAIN PER2DEGS;
BLIMITER PBALIMIT2;
BLIMITER PBALIMIT1;
BINTGR INTEGRATOR;
BGAIN PBAGAIN3;
BSUMJ SUMJ4;
BSUMJ SUMJ3;
BSUMJ SUMJ1;
BGAIN PBAGAIN2;
BGAIN PBAGAIN1;
BGAIN DEG2IN;
BSUMJ SUMJ2;
BVGAIN PBA3MP;
BSOURCE UNITY;
CONNECT ZERO(1) TO PBASWITCH(1);
CONNECT PBASWITCH(1) TO PER2DEGS(1);
CONNECT PBALIMIT2(1) TO PBASWITCH(2);
CONNECT PER2DEGS(1) TO DEG2IN(1);
CONNECT PBAQLIMIT(1) TO PBAGAIN2(1);
CONNECT PBATHETLIMIT(1) TO PBAGAIN1(1);
CONNECT SUMJ3(1) TO SUMJ4(1);
CONNECT PBABIAS(1) TO SUMJ3(2);
CONNECT SUMJ4(1) TO PBAGAIN3(1);
CONNECT PBAGAIN3(1) TO PBALIMIT1(1);
CONNECT PBALIMIT1(1) TO INTEGRATOR(1);
CONNECT INTEGRATOR(1) TO PBALIMIT2(1);
CONNECT PBALIMIT2(1) TO SUMJ4(2);
CONNECT PBAGAIN1(1) TO SUMJ1(2);
CONNECT PBAGAIN2(1) TO SUMJ1(1);
CONNECT SUMJ1(1) TO SUMJ2(1);
CONNECT SUMJ2(1) TO SUMJ3(1);
CONNECT PBA3MP(1) TO SUMJ2(2);
CONNECT UNITY(1) TO PBA3MP(1);


PARENTG
BSOURCE XBTRM;
BGAIN GAIN1;
BGAIN GAIN2;
BTRANSF PRISERVO;
CONNECT SUMJ8(1) TO SUMJ4(2);
CONNECT SUMJ4(1) TO SUMJ5(1);
CONNECT KXBB1S(1) TO SUMJ4(1);
CONNECT SUMJ3(1) TO KXBB1S(1);
CONNECT GAIN2(1) TO SUMJ8(2);
CONNECT GAIN1(1) TO SUMJ8(1);
CONNECT B1SCHK(1) TO SUMJ6(2);
CONNECT SUMJ6(1) TO DEG2RAD(1);
CONNECT BIAS(1) TO SUMJ5(2);
CONNECT XBTRM(1) TO SUMJ1(2);
CONNECT XB(1) TO SUMJ1(1);
CONNECT DEG2RAD(1) TO B1SOUT(1);
CONNECT PitchSASInput_DEG2IN(1) TO SUMJ3(2);
CONNECT PitchAnalogSAS_PALIMIT(1) TO SUMJ7(1);
CONNECT PitchDigitalSAS_PDLIMIT(1) TO SUMJ7(2);
CONNECT SUMJ1(1) TO B1SLIMIT(1);
CONNECT B1SLIMIT(1) TO SUMJ2(2);
CONNECT SUMJ2(1) TO SUMJ3(1);
CONNECT PBA_DEG2IN(1) TO SUMJ2(1);
CONNECT SUMJ5(1) TO PRISERVO(1);
CONNECT PRISERVO(1) TO SUMJ6(1);
CONNECT SUMJ7(1) TO PitchSASInput_PSASACTUATOR(1);


PARENTG
GROUP Lateral
BSOURCE A1SCHK;
BSUMJ SUMJ5;
BSUMJ SUMJ1;
BSOURCE XATRM;
BSOURCE XA;
BLIMITER A1SLIMIT;
BSUMJ SUMJ2;
BGAIN KXAA1S;
BSUMJ SUMJ4;
BGAIN BIAS;
BSINK A1SOUT;
BGAIN DEG2RAD;
GROUP ROLLANALOGSAS
BGAIN RAGROUND;
BGAIN RAGAIN4;
BSUMJ RASUM3;
BDISW ASAS;
BDISW DSAS;
BLIMITER RALIMIT3;
BTRANSF RALAG;
BSUMJ RASUM1;
BGAIN RAGAIN3;
BSUMJ RASUM2;
BGAIN RAGAIN2;
BLIMITER RALIMIT2;
BGAIN RAGAIN1;
BTRANSF RALAG2;
BLIMITER RALIMIT1;
CONNECT RASUM3(1) TO RALIMIT3(1);
CONNECT DSAS(1) TO RASUM3(2);
CONNECT ASAS(1) TO DSAS(2);
CONNECT RAGROUND(1) TO ASAS(1);
CONNECT RAGROUND(1) TO DSAS(1);
CONNECT RAGAIN4(1) TO RASUM3(1);
CONNECT ASAS(1) TO RAGAIN4(1);
CONNECT RAGAIN1(1) TO RALAG(1);
CONNECT RALAG(1) TO RASUM1(2);
CONNECT RASUM1(1) TO RASUM2(1);
CONNECT RASUM2(1) TO ASAS(2);
CONNECT RALIMIT2(1) TO RAGAIN3(1);
CONNECT RAGAIN3(1) TO RALAG2(1);
CONNECT RALAG2(1) TO RASUM2(2);
CONNECT RAGAIN2(1) TO RASUM1(1);
CONNECT RALIMIT1(1) TO RAGAIN2(1);


PARENTG
GROUP ROLLDIGITALSAS
BGAIN RDGROUND;
BGAIN RDGAIN4;
BSUMJ RDSUM3;
BDISW DSAS;
BDISW ASAS;
BLIMITER RDLIMIT2;
BSUMJ RDSUM2;
```

```
BGAIN RDGAIN3;
BSUMJ RDSUM1;
BTRANSF RALAG;
BLIMITER RDLIMIT1;
BGAIN RDGAIN2;
BGAIN RDGAIN1;
CONNECT RDSUM3(1) TO RDLIMIT2(1);
CONNECT RDGAIN4(1) TO RDSUM3(1);
CONNECT DSAS(1) TO RDGAIN4(1);
CONNECT ASAS(1) TO RDSUM3(2);
CONNECT DSAS(1) TO ASAS(2);
CONNECT RDGROUND(1) TO DSAS(1);
CONNECT RDGROUND(1) TO ASAS(1);
CONNECT RDLIMIT1(1) TO RDGAIN3(1);
CONNECT RDSUM1(1) TO RDSUM2(1);
CONNECT RDGAIN3(1) TO RDSUM2(2);
CONNECT RDGAIN2(1) TO RDSUM1(1);
CONNECT RALAG(1) TO RDSUM1(2);
CONNECT RDGAIN1(1) TO RALAG(1);
CONNECT RDSUM2(1) TO DSAS(2);


PARENTG
BSUMJ SUMJ6;
GROUP ROLLSASINPUT
BGAIN DEG2IN;
BGAIN PER2DEG;
BLIMITER RSASLIMIT;
BTRANSF RSASACTUATOR;
CONNECT RSASLIMIT(1) TO PER2DEG(1);
CONNECT PER2DEG(1) TO DEG2IN(1);
CONNECT RSASACTUATOR(1) TO RSASLIMIT(1);


PARENTG
BSUMJ SUMJ3;
BGAIN GAIN1;
BTRANSF PRISERVO;
CONNECT ROLLDIGITALSAS_RDLIMIT2(1) TO SUMJ6(2);
CONNECT A1SCHK(1) TO SUMJ5(2);
CONNECT SUMJ5(1) TO DEG2RAD(1);
CONNECT BIAS(1) TO SUMJ4(2);
CONNECT A1SLIMIT(1) TO SUMJ2(1);
CONNECT SUMJ1(1) TO A1SLIMIT(1);
CONNECT XATRM(1) TO SUMJ1(2);
CONNECT XA(1) TO SUMJ1(1);
CONNECT DEG2RAD(1) TO A1SOUT(1);
CONNECT ROLLSASINPUT_DEG2IN(1) TO SUMJ2(2);
CONNECT ROLLANALOGSAS_RALIMIT3(1) TO SUMJ6(1);
CONNECT SUMJ2(1) TO KXAA1S(1);
CONNECT KXAA1S(1) TO SUMJ3(1);
CONNECT SUMJ3(1) TO SUMJ4(1);
CONNECT GAIN1(1) TO SUMJ3(2);
CONNECT SUMJ4(1) TO PRISERVO(1);
CONNECT PRISERVO(1) TO SUMJ5(1);
CONNECT SUMJ6(1) TO ROLLSASINPUT_RSASACTUATOR(1);


PARENTG
GROUP Directional
BSUMJ SUMJ6;
GROUP YAWSASINPUT
BGAIN DEG2IN;
BGAIN PER2DEG;
BLIMITER YAWSASLIMIT;
BTRANSF YSASACTUATOR;
CONNECT YAWSASLIMIT(1) TO PER2DEG(1);
CONNECT PER2DEG(1) TO DEG2IN(1);
CONNECT YSASACTUATOR(1) TO YAWSASLIMIT(1);


PARENTG
GROUP YAWDIGITALSAS
BSUMJ YDSUM4;
BTRANSF YDWASHOUT;
BGAIN YDGAIN3;
BTRANSF YDLAG2;
BGAIN YDGAIN4;
BGAIN YDGAIN2;
BSOURCE ZERO;
BDISW VX50KTSWITCH1;
BTRANSF YDLAG1;
BSUMJ YDSUM1;
BGAIN YDGAIN1;
BDISW ASAS;
BDISW DSAS;
BSUMJ YDSUM3;
BGAIN YDGAIN5;
BGAIN YDGROUND;
BSUMJ YDSUM2;
BSOURCE YDUNITY;
BSINK YAWSWITCH;
BSUMJ YDSUM6;
BGAIN YDGAIN;
BGAIN VXKNOTS;
BLIMITER YDLIMIT;
CONNECT VX50KTSWITCH1(1) TO YDSUM2(1);
CONNECT ZERO(1) TO VX50KTSWITCH1(1);
CONNECT YDSUM4(1) TO VX50KTSWITCH1(2);
CONNECT YDLAG2(1) TO YDSUM4(1);
CONNECT YDGAIN4(1) TO YDSUM4(2);
CONNECT YDLAG1(1) TO YDSUM1(1);
CONNECT YDGAIN2(1) TO YDLAG1(1);
CONNECT YDWASHOUT(1) TO YDGAIN2(1);
CONNECT YDWASHOUT(1) TO YDGAIN1(1);
CONNECT YDGAIN3(1) TO YDLAG2(1);
CONNECT YDSUM1(1) TO YDSUM2(2);
CONNECT YDGAIN1(1) TO YDSUM1(2);
CONNECT YDGROUND(1) TO ASAS(1);
CONNECT YDGROUND(1) TO DSAS(1);
CONNECT DSAS(1) TO ASAS(2);
CONNECT ASAS(1) TO YDSUM3(2);
CONNECT DSAS(1) TO YDGAIN5(1);
CONNECT YDGAIN5(1) TO YDSUM3(1);
CONNECT YDSUM2(1) TO DSAS(2);
CONNECT YDUNITY(1) TO YDGAIN(1);
CONNECT YDSUM6(1) TO YAWSWITCH(1);
CONNECT VXKNOTS(1) TO YDSUM6(2);
CONNECT YDGAIN(1) TO YDSUM6(1);
CONNECT YDSUM3(1) TO YDLIMIT(1);


PARENTG
BSOURCE THRCHK;
BSUMJ SUMJ5;
BSUMJ SUMJ1;
BSOURCE XPTRM;
BSOURCE XP;
BLIMITER THETTRLIMIT;
BSUMJ SUMJ2;
BGAIN KXPTHR;
BSUMJ SUMJ4;
BGAIN BIAS;
BSINK THETTROUT;
BGAIN DEG2RAD;
GROUP YAWANALOGSAS
BTRANSF YAWASHOUT;
BSOURCE ZERO;
BGAIN YAGAIN4;
BSUMJ YASUM2;
BDISW VX50KTSWITCH2;
BGAIN YAGAIN3;
BDISW VX50KTSWITCH1;
BLIMITER YALIMIT3;
BDISW DSAS;
BDISW ASAS;
BSUMJ YASUM3;
BGAIN YAGAIN5;
BGAIN YAGROUND;
BSUMJ YASUM1;
BGAIN YAGAIN2;
BGAIN YAGAIN1;
BTRANSF YALAG;
```

```
BLIMITER YALIMIT1;
BLIMITER YALIMIT2;
CONNECT ZERO(1) TO VX50KTSWITCH2(1);
CONNECT VX50KTSWITCH2(1) TO YASUM2(2);
CONNECT YASUM2(1) TO ASAS(2);
CONNECT YASUM1(1) TO YASUM2(1);
CONNECT YALAG(1) TO YASUM1(2);
CONNECT YAGAIN1(1) TO YASUM1(1);
CONNECT YAGAIN4(1) TO VX50KTSWITCH2(2);
CONNECT VX50KTSWITCH1(1) TO YALAG(1);
CONNECT YAGAIN2(1) TO VX50KTSWITCH1(1);
CONNECT YAGAIN3(1) TO VX50KTSWITCH1(2);
CONNECT YALIMIT2(1) TO YAGAIN2(1);
CONNECT YAGROUND(1) TO DSAS(1);
CONNECT YAGROUND(1) TO ASAS(1);
CONNECT ASAS(1) TO DSAS(2);
CONNECT DSAS(1) TO YASUM3(2);
CONNECT ASAS(1) TO YAGAIN5(1);
CONNECT YAGAIN5(1) TO YASUM3(1);
CONNECT YASUM3(1) TO YALIMIT3(1);
CONNECT YAWASHOUT(1) TO YALIMIT1(1);
CONNECT YAWASHOUT(1) TO YALIMIT2(1);
CONNECT YALIMIT1(1) TO YAGAIN1(1);


PARENTG
BSUMJ SUMJ3;
BGAIN GAIN1;
BTRANSF PRISERVO;
CONNECT THRCHK(1) TO SUMJ5(2);
CONNECT SUMJ5(1) TO DEG2RAD(1);
CONNECT YAWSASINPUT_DEG2IN(1) TO SUMJ2(2);
CONNECT YAWANALOGSAS_YALIMIT3(1) TO SUMJ6(1);
CONNECT BIAS(1) TO SUMJ4(2);
CONNECT THETTRLIMIT(1) TO SUMJ2(1);
CONNECT SUMJ1(1) TO THETTRLIMIT(1);
CONNECT XP(1) TO SUMJ1(1);
CONNECT XPTRM(1) TO SUMJ1(2);
CONNECT DEG2RAD(1) TO THETTROUT(1);
CONNECT SUMJ2(1) TO KXPTHR(1);
CONNECT KXPTHR(1) TO SUMJ3(1);
CONNECT SUMJ3(1) TO SUMJ4(1);
CONNECT GAIN1(1) TO SUMJ3(2);
CONNECT SUMJ4(1) TO PRISERVO(1);
CONNECT PRISERVO(1) TO SUMJ5(1);
CONNECT SUMJ6(1) TO YAWSASINPUT_YSASACTUATOR(1);
CONNECT YAWDIGITALSAS_YDLIMIT(1) TO SUMJ6(2);


PARENTG
CONNECT Collective_KXCTH0(1) TO Longitudinal_GAIN1(1);
CONNECT Collective_KXCTH0(1) TO Lateral_GAIN1(1);
CONNECT Collective_KXCTH0(1) TO Directional_GAIN1(1);
CONNECT Directional_KXPTHR(1) TO Longitudinal_GAIN2(1);


PARENTG
CONNECT FlightControls_Collective_THETLIMIT(1) TO Stabilator_XCPC(1);
CONNECT Sensors_RSENSOR(1) TO
FlightControls_Directional_YAWDIGITALSAS_YDWASHOUT(1);
CONNECT Sensors_RSENSOR(1) TO
FlightControls_Directional_YAWANALOGSAS_YAWASHOUT(1);
CONNECT Sensors_LATACCEL(1) TO
FlightControls_Directional_YAWDIGITALSAS_YDGAIN3(1);
CONNECT Sensors_PSENSOR(1) TO
FlightControls_Directional_YAWDIGITALSAS_YDGAIN4(1);
CONNECT Sensors_PSENSOR(1) TO
FlightControls_Directional_YAWANALOGSAS_YAGAIN4(1);
CONNECT Sensors_LATACCEL(1) TO
FlightControls_Directional_YAWANALOGSAS_YAGAIN3(1);
CONNECT Sensors_LATACCEL(1) TO Stabilator_KTS2FPS2(1);
CONNECT Sensors_UNITY(1) TO
FlightControls_Lateral_ROLLANALOGSAS_RAGROUND(1);
CONNECT Sensors_UNITY(1) TO
FlightControls_Lateral_ROLLDIGITALSAS_RDGROUND(1);
CONNECT Sensors_UNITY(1) TO
FlightControls_Longitudinal_PitchDigitalSAS_PDGROUND(1);
CONNECT Sensors_UNITY(1) TO
FlightControls_Longitudinal_PitchAnalogSAS_PAGROUND(1);
CONNECT Sensors_UKNOTS(1) TO Stabilator_VXLAG(1);
CONNECT Sensors_QSENSOR(1) TO Stabilator_GAIN1(1);
CONNECT Sensors_UNITY(1) TO FlightControls_Longitudinal_BIAS(1);
CONNECT Sensors_UNITY(1) TO FlightControls_Lateral_BIAS(1);
CONNECT Sensors_UNITY(1) TO FlightControls_Collective_BIAS(1);
CONNECT Sensors_UNITY(1) TO FlightControls_Directional_BIAS(1);
CONNECT Sensors_thetadeg(1) TO
FlightControls_Longitudinal_PBA_PBATHETLIMIT(1);
CONNECT Sensors_UNITY(1) TO FlightControls_Longitudinal_PBA_PBABIAS(1);
CONNECT Sensors_QSENSOR(1) TO
FlightControls_Longitudinal_PBA_PBAQLIMIT(1);
CONNECT Sensors_UNITY(1) TO
FlightControls_Directional_YAWANALOGSAS_YAGROUND(1);
CONNECT Sensors_UNITY(1) TO
FlightControls_Directional_YAWDIGITALSAS_YDGROUND(1);
CONNECT Sensors_PSENSOR(1) TO
FlightControls_Lateral_ROLLANALOGSAS_RAGAIN1(1);
CONNECT Sensors_phideg(1) TO
FlightControls_Lateral_ROLLANALOGSAS_RALIMIT2(1);
CONNECT Sensors_PSENSOR(1) TO
FlightControls_Lateral_ROLLDIGITALSAS_RDGAIN1(1);
CONNECT Sensors_PSENSOR(1) TO
FlightControls_Lateral_ROLLDIGITALSAS_RDGAIN2(1);
```

```
CONNECT Sensors_phideg(1) TO
FlightControls_Lateral_ROLLDIGITALSAS_RDLIMIT1(1);
CONNECT Sensors_QSENSOR(1) TO
FlightControls_Longitudinal_PitchAnalogSAS_PAWASHOUT(1);
CONNECT Sensors_QSENSOR(1) TO
FlightControls_Longitudinal_PitchDigitalSAS_PDWASHOUT(1);
CONNECT Sensors_PSENSOR(1) TO
FlightControls_Lateral_ROLLANALOGSAS_RALIMIT1(1);
CONNECT Sensors_UKNOTS(1) TO
FlightControls_Directional_YAWDIGITALSAS_VXKNOTS(1);


PARENTG


PARENTG


GOTO MODEL
GOTO CONTROL
GOTO Sensors
      VDOTLIMIT_N =1;
      VDOTLIMIT_LL      =-16.085;
      VDOTLIMIT_UL      =16.085;

      RLIMIT_N    =1;
      RLIMIT_LL   =-40.0;
      RLIMIT_UL   =40.0;

      QLIMIT_N    =1;
      QLIMIT_LL   =-40.0;
      QLIMIT_UL   =40.0;

      PLIMIT_N    =1;
      PLIMIT_LL   =-40.0;
      PLIMIT_UL   =40.0;

      DOFSENSORA_N      =9;
      DOFSENSORA_U      =&WORLD_DATA_PFRASENS;

      psi_NI      =9;
      psi_NO      =1;
      psi_K =[0 0 1 0 0 0 0 0 0];

      theta_NI    =9;
      theta_NO    =1;
      theta_K     =[0 1 0 0 0 0 0 0 0];

      phi_NI      =9;
      phi_NO      =1;
      phi_K =[1 0 0 0 0 0 0 0 0];
```

```
r_NI  =9;
r_NO  =1;
r_K   =[0 0 0 0 0 1 0 0 0];

q_NI  =9;
q_NO  =1;
q_K   =[0 0 0 0 1 0 0 0 0];

p_NI  =9;
p_NO  =1;
p_K   =[0 0 0 1 0 0 0 0 0 ];

psideg_N     =1;
psideg_K     =&WORLD_DATA_R2D;

thetadeg_N   =1;
thetadeg_K   =&WORLD_DATA_R2D;

phideg_N     =1;
phideg_K     =&WORLD_DATA_R2D;

rdeg_N       =1;
rdeg_K       =&WORLD_DATA_R2D;

qdeg_N       =1;
qdeg_K       =&WORLD_DATA_R2D;

pdeg_N       =1;
pdeg_K       =&WORLD_DATA_R2D;

ALPHASRC_N   =1;
ALPHASRC_U   =&WORLD_DATA_ALFWF;

BETASRC_N    =1;
BETASRC_U    =&WORLD_DATA_BETWF;

UNITY_N      =1;
UNITY_U      =1;

DOFSENSORL_N       =9;
DOFSENSORL_U       =&WORLD_DATA_PFRLSENS;

U_NI  =9;
U_NO  =1;
U_K   =[0 0 0 1 0 0 0 0 0];

V_NI  =9;
V_NO  =1;
V_K   =[0 0 0 0 1 0 0 0 0];

W_NI  =9;
```

```
        W_NO  =1;
        W_K   =[0 0 0 0 0 1 0 0 0];

        VDOT_NI      =9;
        VDOT_NO      =1;
        VDOT_K       =[0 0 0 0 0 0 0 1 0];

        UKNOTS_N     =1;
        UKNOTS_K     =&WORLD_MODEL_CONTROL_DATA_F2K;

        VKNOTS_N     =1;
        VKNOTS_K     =&WORLD_MODEL_CONTROL_DATA_F2K;

        WKNOTS_N     =1;
        WKNOTS_K     =&WORLD_MODEL_CONTROL_DATA_F2K;

        PSENSOR_N    =2;
        PSENSOR_NUM =&WORLD_MODEL_CONTROL_DATA_SNUM;
        PSENSOR_DEN =&WORLD_MODEL_CONTROL_DATA_SDEN;

        QSENSOR_N    =2;
        QSENSOR_NUM =&WORLD_MODEL_CONTROL_DATA_SNUM;
        QSENSOR_DEN =&WORLD_MODEL_CONTROL_DATA_SDEN;

        RSENSOR_N    =2;
        RSENSOR_NUM =&WORLD_MODEL_CONTROL_DATA_SNUM;
        RSENSOR_DEN =&WORLD_MODEL_CONTROL_DATA_SDEN;

        ALPHA_N      =1;

        BETA_N       =1;

        ALPHADELAY_N       =1;
        ALPHADELAY_NUM     =[0 1];
        ALPHADELAY_DEN     =[.0001 1];

        BETADELAY_N =1;
        BETADELAY_NUM      =[0 1];
        BETADELAY_DEN      =[.0001 1];

        LATACCEL_N  =2;
        LATACCEL_NUM       =&WORLD_MODEL_CONTROL_DATA_LATACELNUM;
        LATACCEL_DEN       =&WORLD_MODEL_CONTROL_DATA_LATACELDEN;


PARENTG
GOTO Stabilator
        KTS2FPS2_N  =1;
        KTS2FPS2_K  =1;
```

```
DEG2RAD_NI   =1;
DEG2RAD_NO   =3;
DEG2RAD_K    =[.01745 0 0]';

STABINPUTSINK_N    =2;

STABINPUT_NI1      =1;
STABINPUT_NI2      =1;
STABINPUT_NO       =2;
STABINPUT_SM       =[1 0;0 1];

VXLAG_N      =1;
VXLAG_NUM    =[0 1];
VXLAG_DEN    =[.398 1];

STABPOSITMAP_NI    =1;
STABPOSITMAP_NO    =1;
STABPOSITMAP_NPARS       =2;
STABPOSITMAP_BPT   =[ 0 180 10 19 ;0 100 10 11]';
STABPOSITMAP_NTAB =209;
STABPOSITMAP_NROWS       =1;
STABPOSITMAP_KTAB =&WORLD_MODEL_CONTROL_DATA_STABPOSIT;
STABPOSITMAP_VAR   =[1 1]'   //first input is VXLAG;

UNITY_N      =1;
UNITY_U      =1;

GAINMAP_NI  =1;
GAINMAP_NO  =1;
GAINMAP_NPARS      =1;
GAINMAP_BPT =[0 180 10 19]';
GAINMAP_NTAB       =19;
GAINMAP_NROWS      =1;
GAINMAP_KTAB       =&WORLD_MODEL_CONTROL_DATA_GAINMAP;
GAINMAP_VAR =1;

PROD1_N      =1;

SUMJ1_N      =1;
SUMJ1_SA1    =1;
SUMJ1_SA2    =1;

STABPOSIT_N =3;

GAIN1_N      =1;
GAIN1_K      =.1526;

SUMJ2_N      =1;
SUMJ2_SA1    =1;
SUMJ2_SA2    =1;
```

```
    RANGELIMIT_N      =1;
    RANGELIMIT_LL     =-8;
    RANGELIMIT_UL     =40;

    ACTUATOR_N  =2;
    ACTUATOR_NUM      =[0 0 1];
    ACTUATOR_DEN      =[.0254 .223 1];

    VECTORUNITY_N     =1;
    VECTORUNITY_U     =1;

    XCPC_N      =1;
    XCPC_K      =10;



PARENTG
GOTO FlightControls
GOTO Collective
    THETLIMIT_N =1;
    THETLIMIT_LL      =&WORLD_MODEL_CONTROL_DATA_THETMRLL;
    THETLIMIT_UL      =&WORLD_MODEL_CONTROL_DATA_THETMRUL;

    TH0CHK_N    =1;
    TH0CHK_U    =&WORLD_MODEL_CONTROL_DATA_TH0CHK;

    SUMJ4_N     =1;
    SUMJ4_SA1   =1;
    SUMJ4_SA2   =1;

    THETA0OUT_N =1;

    BIAS_N      =1;
    BIAS_K      =&WORLD_MODEL_CONTROL_DATA_THETMRBIAS;

    SUMJ3_N     =1;
    SUMJ3_SA1   =1;
    SUMJ3_SA2   =1;

    KXCTH0_N    =1;
    KXCTH0_K    =&WORLD_MODEL_CONTROL_DATA_KXCTH0;

    XCILS   _N  =1;
    XCILS   _U  =&WORLD_MODEL_CONTROL_DATA_XCS;

    SUMJ2_N     =1;
    SUMJ2_SA1   =1;
    SUMJ2_SA2   =1;

    SUMJ1_N     =1;
    SUMJ1_SA1   =1;
```

```
      SUMJ1_SA2    =1;

      XC_N  =1;
      XC_U  =&WORLD_MODEL_CONTROL_DATA_XC;

      XCTRM_N      =1;
      XCTRM_U      =&WORLD_MODEL_CONTROL_DATA_XCTRM;

      DEG2RAD_N    =1;
      DEG2RAD_K    =&WORLD_DATA_D2R;

      PRISERVO_N   =2;
      PRISERVO_NUM       =&WORLD_MODEL_CONTROL_DATA_PRISERVONUM;
      PRISERVO_DEN       =&WORLD_MODEL_CONTROL_DATA_PRISERVODEN;


PARENTG
GOTO Longitudinal
      SUMJ4_N      =1;
      SUMJ4_SA1    =1;
      SUMJ4_SA2    =1;

      SUMJ8_N      =1;
      SUMJ8_SA1    =1;
      SUMJ8_SA2    =1;

      B1SCHK_N     =1;
      B1SCHK_U     =&WORLD_MODEL_CONTROL_DATA_B1SCHK;

      SUMJ6_N      =1;
      SUMJ6_SA1    =1;
      SUMJ6_SA2    =1;

      XB_N  =1;
      XB_U  =&WORLD_MODEL_CONTROL_DATA_XB;

      SUMJ1_N      =1;
      SUMJ1_SA1    =1;
      SUMJ1_SA2    =1;

      B1SLIMIT_N   =1;
      B1SLIMIT_LL =&WORLD_MODEL_CONTROL_DATA_B1SLL;
      B1SLIMIT_UL =&WORLD_MODEL_CONTROL_DATA_B1SUL;

      SUMJ3_N      =1;
      SUMJ3_SA1    =1;
      SUMJ3_SA2    =1;

      KXBB1S_N     =1;
      KXBB1S_K     =&WORLD_MODEL_CONTROL_DATA_KXBB1S;
```

```
        SUMJ5_N      =1;
        SUMJ5_SA1    =1;
        SUMJ5_SA2    =1;

        BIAS_N       =1;
        BIAS_K       =&WORLD_MODEL_CONTROL_DATA_B1SBIAS;

        B1SOUT_N     =1;

        DEG2RAD_N    =1;
        DEG2RAD_K    =&WORLD_DATA_D2R;

        SUMJ7_N      =1;
        SUMJ7_SA1    =1;
        SUMJ7_SA2    =1;

GOTO PitchAnalogSAS
        PAGROUND_N  =1;
        PAGROUND_K  =0;

        DSAS_N       =1;
        DSAS_SW      =&WORLD_MODEL_CONTROL_DATA_DSAS;

        ASAS_N       =1;
        ASAS_SW      =&WORLD_MODEL_CONTROL_DATA_ASAS;

        PASUM2_N     =1;
        PASUM2_SA1   =1;
        PASUM2_SA2   =-1;

        PAGAIN3_N    =1;
        PAGAIN3_K    =2;

        PALIMIT_N    =1;
        PALIMIT_LL   =&WORLD_MODEL_CONTROL_DATA_PALLIMIT;
        PALIMIT_UL   =&WORLD_MODEL_CONTROL_DATA_PAULIMIT;

        PALAG_N      =1;
        PALAG_NUM    =&WORLD_MODEL_CONTROL_DATA_PALAGNUM;
        PALAG_DEN    =&WORLD_MODEL_CONTROL_DATA_PALAGDEN;

        PASUM1_N     =1;
        PASUM1_SA1   =1;
        PASUM1_SA2   =1;

        PAGAIN1_N    =1;
        PAGAIN1_K    =&WORLD_MODEL_CONTROL_DATA_PAGAIN1;

        PAGAIN2_N    =1;
        PAGAIN2_K    =&WORLD_MODEL_CONTROL_DATA_PAGAIN2;
```

```
      PAWASHOUT_N =1;
      PAWASHOUT_NUM      =&WORLD_MODEL_CONTROL_DATA_PAWOUTNUM;
      PAWASHOUT_DEN      =&WORLD_MODEL_CONTROL_DATA_PAWOUTDEN;


PARENTG
GOTO PitchDigitalSAS
      PDGROUND_N  =1;
      PDGROUND_K  =0;

      ASAS_N      =1;
      ASAS_SW     =&WORLD_MODEL_CONTROL_DATA_ASAS;

      DSAS_N      =1;
      DSAS_SW     =&WORLD_MODEL_CONTROL_DATA_DSAS;

      PDSUM2_N    =1;
      PDSUM2_SA1  =1;
      PDSUM2_SA2  =-1;

      PDGAIN3_N   =1;
      PDGAIN3_K   =2;

      PDLIMIT_N   =1;
      PDLIMIT_LL  =&WORLD_MODEL_CONTROL_DATA_PDLLIMIT;
      PDLIMIT_UL  =&WORLD_MODEL_CONTROL_DATA_PDULIMIT;

      PDSUM1_N    =1;
      PDSUM1_SA1  =1;
      PDSUM1_SA2  =1;

      PDLAG_N     =1;
      PDLAG_NUM   =&WORLD_MODEL_CONTROL_DATA_PDLAGNUM;
      PDLAG_DEN   =&WORLD_MODEL_CONTROL_DATA_PDLAGDEN;

      PDGAIN1_N   =1;
      PDGAIN1_K   =&WORLD_MODEL_CONTROL_DATA_PDGAIN1;

      PDGAIN2_N   =1;
      PDGAIN2_K   =&WORLD_MODEL_CONTROL_DATA_PDGAIN2;

      PDWASHOUT_N =1;
      PDWASHOUT_NUM      =&WORLD_MODEL_CONTROL_DATA_PDWOUTNUM;
      PDWASHOUT_DEN      =&WORLD_MODEL_CONTROL_DATA_PDWOUTDEN;


PARENTG
GOTO PitchSASInput
      DEG2IN_N    =1;
```

```
        DEG2IN_K     =&WORLD_MODEL_CONTROL_DATA_KXBB1SINV;

        PER2DEG_N    =1;
        PER2DEG_K    =&WORLD_MODEL_CONTROL_DATA_KPCDGP;

        PSASLIMIT_N =1;
        PSASLIMIT_LL       =&WORLD_MODEL_CONTROL_DATA_PSASLLIMIT;
        PSASLIMIT_UL       =&WORLD_MODEL_CONTROL_DATA_PSASULIMIT;

        PSASACTUATOR_N     =2;
        PSASACTUATOR_NUM   =&WORLD_MODEL_CONTROL_DATA_PSASACTNUM;
        PSASACTUATOR_DEN   =&WORLD_MODEL_CONTROL_DATA_PSASACTDEN;



PARENTG
        SUMJ2_N      =1;
        SUMJ2_SA1    =1;
        SUMJ2_SA2    =1;

GOTO PBA
        ZERO_N       =1;
        ZERO_U       =0;

        PBASWITCH_N =1;
        PBASWITCH_SW       =&WORLD_MODEL_CONTROL_DATA_PBASW;

        PBABIAS_N    =1;
        PBABIAS_K    =&WORLD_MODEL_CONTROL_DATA_PBABIAS;

        PBATHETLIMIT_N     =1;
        PBATHETLIMIT_LL    =&WORLD_MODEL_CONTROL_DATA_PBATHETLLIMIT;
        PBATHETLIMIT_UL    =&WORLD_MODEL_CONTROL_DATA_PBATHETULIMIT;

        PBAQLIMIT_N =1;
        PBAQLIMIT_LL       =&WORLD_MODEL_CONTROL_DATA_PBAQLLIMIT;
        PBAQLIMIT_UL       =&WORLD_MODEL_CONTROL_DATA_PBAQULIMIT;

        PER2DEGS_N   =1;
        PER2DEGS_K   =&WORLD_MODEL_CONTROL_DATA_KPCDGP;

        PBALIMIT2_N =1;
        PBALIMIT2_LL       =&WORLD_MODEL_CONTROL_DATA_PBALLIMIT2;
        PBALIMIT2_UL       =&WORLD_MODEL_CONTROL_DATA_PBAULIMIT2;

        PBALIMIT1_N =1;
        PBALIMIT1_LL       =&WORLD_MODEL_CONTROL_DATA_PBALLIMIT1;
        PBALIMIT1_UL       =&WORLD_MODEL_CONTROL_DATA_PBAULIMIT1;

        INTEGRATOR_N       =1;
        PBAGAIN3_N   =1;
        PBAGAIN3_K   =&WORLD_MODEL_CONTROL_DATA_PBAGAIN3;

        SUMJ4_N      =1;
        SUMJ4_SA1    =1;
        SUMJ4_SA2    =-1;

        SUMJ3_N      =1;
        SUMJ3_SA1    =1;
        SUMJ3_SA2    =1;

        SUMJ1_N      =1;
        SUMJ1_SA1    =1;
        SUMJ1_SA2    =1;

        PBAGAIN2_N   =1;
        PBAGAIN2_K   =&WORLD_MODEL_CONTROL_DATA_PBAGAIN2;

        PBAGAIN1_N   =1;
        PBAGAIN1_K   =&WORLD_MODEL_CONTROL_DATA_PBAGAIN1;

        DEG2IN_N     =1;
        DEG2IN_K     =&WORLD_MODEL_CONTROL_DATA_KXBB1SINV;

        SUMJ2_N      =1;
        SUMJ2_SA1    =1;
        SUMJ2_SA2    =1;

        PBA3MP_NI    =1;
        PBA3MP_NO    =1;
        PBA3MP_NPARS       =1;
        PBA3MP_BPT   =[0 200 10 21]';
        PBA3MP_NTAB =21;
        PBA3MP_NROWS       =1;
        PBA3MP_KTAB =&WORLD_MODEL_CONTROL_DATA_PBAMAP;
        PBA3MP_VAR   =1;

        UNITY_N      =1;
        UNITY_U      =1;


PARENTG
        XBTRM_N      =1;
        XBTRM_U      =&WORLD_MODEL_CONTROL_DATA_XBTRM;

        GAIN1_N      =1;
        GAIN1_K      =.29;

        GAIN2_N      =1;
        GAIN2_K      =-.2936;
```

```
      PRISERVO_N  =2;
      PRISERVO_NUM       =&WORLD_MODEL_CONTROL_DATA_PRISERVONUM;
      PRISERVO_DEN       =&WORLD_MODEL_CONTROL_DATA_PRISERVODEN;


PARENTG
GOTO Lateral
      A1SCHK_N    =1;
      A1SCHK_U    =&WORLD_MODEL_CONTROL_DATA_A1SCHK;

      SUMJ5_N     =1;
      SUMJ5_SA1   =1;
      SUMJ5_SA2   =1;

      SUMJ1_N     =1;
      SUMJ1_SA1   =1;
      SUMJ1_SA2   =1;

      XATRM_N     =1;
      XATRM_U     =&WORLD_MODEL_CONTROL_DATA_XATRM;

      XA_N  =1;
      XA_U  =&WORLD_MODEL_CONTROL_DATA_XA;

      A1SLIMIT_N  =1;
      A1SLIMIT_LL =0;
      A1SLIMIT_UL =10.0;

      SUMJ2_N     =1;
      SUMJ2_SA1   =1;
      SUMJ2_SA2   =1;

      KXAA1S_N    =1;
      KXAA1S_K    =&WORLD_MODEL_CONTROL_DATA_KXAA1S;

      SUMJ4_N     =1;
      SUMJ4_SA1   =1;
      SUMJ4_SA2   =1;

      BIAS_N      =1;
      BIAS_K      =&WORLD_MODEL_CONTROL_DATA_A1SBIAS;

      A1SOUT_N    =1;

      DEG2RAD_N   =1;
      DEG2RAD_K   =&WORLD_DATA_D2R;

GOTO ROLLANALOGSAS
      RAGROUND_N  =1;
      RAGROUND_K  =0;
```

```
RAGAIN4_N    =1;
RAGAIN4_K    =2;

RASUM3_N     =1;
RASUM3_SA1   =1;
RASUM3_SA2   =-1;

ASAS_N       =1;
ASAS_SW      =&WORLD_MODEL_CONTROL_DATA_ASAS;

DSAS_N       =1;
DSAS_SW      =&WORLD_MODEL_CONTROL_DATA_DSAS;

RALIMIT3_N   =1;
RALIMIT3_LL =&WORLD_MODEL_CONTROL_DATA_RALLIMIT3;
RALIMIT3_UL =&WORLD_MODEL_CONTROL_DATA_RAULIMIT3;

RALAG_N      =1;
RALAG_NUM    =&WORLD_MODEL_CONTROL_DATA_RALAGNUM;
RALAG_DEN    =&WORLD_MODEL_CONTROL_DATA_RALAGDEN;

RASUM1_N     =1;
RASUM1_SA1   =1;
RASUM1_SA2   =1;

RAGAIN3_N    =1;
RAGAIN3_K    =&WORLD_MODEL_CONTROL_DATA_RAGAIN3;

RASUM2_N     =1;
RASUM2_SA1   =1;
RASUM2_SA2   =1;

RAGAIN2_N    =1;
RAGAIN2_K    =&WORLD_MODEL_CONTROL_DATA_RAGAIN2;

RALIMIT2_N   =1;
RALIMIT2_LL =&WORLD_MODEL_CONTROL_DATA_RALLIMIT2;
RALIMIT2_UL =&WORLD_MODEL_CONTROL_DATA_RAULIMIT2;

RAGAIN1_N    =1;
RAGAIN1_K    =&WORLD_MODEL_CONTROL_DATA_RAGAIN1;

RALAG2_N     =1;
RALAG2_NUM   =&WORLD_MODEL_CONTROL_DATA_RALAG2NUM;
RALAG2_DEN   =&WORLD_MODEL_CONTROL_DATA_RALAG2DEN;

RALIMIT1_N   =1;
RALIMIT1_LL =&WORLD_MODEL_CONTROL_DATA_RALLIMIT1;
RALIMIT1_UL =&WORLD_MODEL_CONTROL_DATA_RAULIMIT1;
```

```
PARENTG
GOTO ROLLDIGITALSAS
      RDGROUND_N  =1;
      RDGROUND_K  =0;

      RDGAIN4_N   =1;
      RDGAIN4_K   =2;

      RDSUM3_N    =1;
      RDSUM3_SA1  =1;
      RDSUM3_SA2  =-1;

      DSAS_N      =1;
      DSAS_SW     =&WORLD_MODEL_CONTROL_DATA_DSAS;

      ASAS_N      =1;
      ASAS_SW     =&WORLD_MODEL_CONTROL_DATA_ASAS;

      RDLIMIT2_N  =1;
      RDLIMIT2_LL =&WORLD_MODEL_CONTROL_DATA_RDLLIMIT2;
      RDLIMIT2_UL =&WORLD_MODEL_CONTROL_DATA_RDULIMIT2;

      RDSUM2_N    =1;
      RDSUM2_SA1  =1;
      RDSUM2_SA2  =1;

      RDGAIN3_N   =1;
      RDGAIN3_K   =&WORLD_MODEL_CONTROL_DATA_RDGAIN3;

      RDSUM1_N    =1;
      RDSUM1_SA1  =1;
      RDSUM1_SA2  =1;

      RALAG_N     =1;
      RALAG_NUM   =[0 1];
      RALAG_DEN   =[1 1];

      RDLIMIT1_N  =1;
      RDLIMIT1_LL =&WORLD_MODEL_CONTROL_DATA_RDLLIMIT1;
      RDLIMIT1_UL =&WORLD_MODEL_CONTROL_DATA_RDULIMIT1;

      RDGAIN2_N   =1;
      RDGAIN2_K   =&WORLD_MODEL_CONTROL_DATA_RDGAIN2;

      RDGAIN1_N   =1;
      RDGAIN1_K   =&WORLD_MODEL_CONTROL_DATA_RDGAIN1;


PARENTG
      SUMJ6_N     =1;
      SUMJ6_SA1    =1;
      SUMJ6_SA2    =1;

GOTO ROLLSASINPUT
      DEG2IN_N     =1;
      DEG2IN_K     =&WORLD_MODEL_CONTROL_DATA_KXAA1SINV;

      PER2DEG_N    =1;
      PER2DEG_K    =&WORLD_MODEL_CONTROL_DATA_KPCDGR;

      RSASLIMIT_N =1;
      RSASLIMIT_LL       =&WORLD_MODEL_CONTROL_DATA_RSASLLIMIT;
      RSASLIMIT_UL       =&WORLD_MODEL_CONTROL_DATA_RSASULIMIT;

      RSASACTUATOR_N     =2;
      RSASACTUATOR_NUM   =&WORLD_MODEL_CONTROL_DATA_RSASACTNUM;
      RSASACTUATOR_DEN   =&WORLD_MODEL_CONTROL_DATA_RSASACTDEN;



PARENTG
      SUMJ3_N      =1;
      SUMJ3_SA1    =1;
      SUMJ3_SA2    =1;

      GAIN1_N      =1;
      GAIN1_K      =-.16;

      PRISERVO_N  =2;
      PRISERVO_NUM       =&WORLD_MODEL_CONTROL_DATA_PRISERVONUM;
      PRISERVO_DEN       =&WORLD_MODEL_CONTROL_DATA_PRISERVODEN;



PARENTG
GOTO Directional
      SUMJ6_N      =1;
      SUMJ6_SA1    =1;
      SUMJ6_SA2    =1;

GOTO YAWSASINPUT
      DEG2IN_N     =1;
      DEG2IN_K     =&WORLD_MODEL_CONTROL_DATA_KXPTHRINV;

      PER2DEG_N    =1;
      PER2DEG_K    =&WORLD_MODEL_CONTROL_DATA_KPCDGY;

      YAWSASLIMIT_N      =1;
      YAWSASLIMIT_LL     =-10.0;
      YAWSASLIMIT_UL     =10.0;
```

```
        YSASACTUATOR_N      =2;
        YSASACTUATOR_NUM    =&WORLD_MODEL_CONTROL_DATA_YSASACTNUM;
        YSASACTUATOR_DEN    =&WORLD_MODEL_CONTROL_DATA_YSASACTDEN;



PARENTG
GOTO YAWDIGITALSAS
        YDSUM4_N     =1;
        YDSUM4_SA1   =1;
        YDSUM4_SA2   =1;

        YDWASHOUT_N =1;
        YDWASHOUT_NUM       =&WORLD_MODEL_CONTROL_DATA_YDWOUTNUM;
        YDWASHOUT_DEN       =&WORLD_MODEL_CONTROL_DATA_YDWOUTDEN;

        YDGAIN3_N    =1;
        YDGAIN3_K    =&WORLD_MODEL_CONTROL_DATA_YDGAIN3;

        YDLAG2_N     =1;
        YDLAG2_NUM   =&WORLD_MODEL_CONTROL_DATA_YDLAG2NUM;
        YDLAG2_DEN   =&WORLD_MODEL_CONTROL_DATA_YDLAG2DEN;

        YDGAIN4_N    =1;
        YDGAIN4_K    =&WORLD_MODEL_CONTROL_DATA_YDGAIN4;

        YDGAIN2_N    =1;
        YDGAIN2_K    =&WORLD_MODEL_CONTROL_DATA_YDGAIN2;

        ZERO_N       =1;
        ZERO_U       =0;

        VX50KTSWITCH1_N    =1;
        VX50KTSWITCH1_SW   =&WORLD_MODEL_CONTROL_DATA_V50KTS;

        YDLAG1_N     =1;
        YDLAG1_NUM   =&WORLD_MODEL_CONTROL_DATA_YDLAG1NUM;
        YDLAG1_DEN   =&WORLD_MODEL_CONTROL_DATA_YDLAG1DEN;

        YDSUM1_N     =1;
        YDSUM1_SA1   =1;
        YDSUM1_SA2   =1;

        YDGAIN1_N    =1;
        YDGAIN1_K    =&WORLD_MODEL_CONTROL_DATA_YDGAIN1;

        ASAS_N       =1;
        ASAS_SW      =&WORLD_MODEL_CONTROL_DATA_ASAS;

        DSAS_N       =1;
        DSAS_SW      =&WORLD_MODEL_CONTROL_DATA_DSAS;
        YDSUM3_N     =1;
        YDSUM3_SA1   =1;
        YDSUM3_SA2   =-1;

        YDGAIN5_N    =1;
        YDGAIN5_K    =2;

        YDGROUND_N   =1;
        YDGROUND_K   =0;

        YDSUM2_N     =1;
        YDSUM2_SA1   =1;
        YDSUM2_SA2   =1;

        YDUNITY_N    =1;
        YDUNITY_U    =1;

        YAWSWITCH_N =1;

        YDSUM6_N     =1;
        YDSUM6_SA1   =-1;
        YDSUM6_SA2   =1;

        YDGAIN_N     =1;
        YDGAIN_K     =49.0;

        VXKNOTS_N    =1;
        VXKNOTS_K    =1;

        YDLIMIT_N    =1;
        YDLIMIT_LL   =&WORLD_MODEL_CONTROL_DATA_YDLLIMIT;
        YDLIMIT_UL   =&WORLD_MODEL_CONTROL_DATA_YDULIMIT;


PARENTG
        THRCHK_N     =1;
        THRCHK_U     =&WORLD_MODEL_CONTROL_DATA_THRCHK;

        SUMJ5_N      =1;
        SUMJ5_SA1    =1;
        SUMJ5_SA2    =1;

        SUMJ1_N      =1;
        SUMJ1_SA1    =1;
        SUMJ1_SA2    =1;

        XPTRM_N      =1;
        XPTRM_U      =&WORLD_MODEL_CONTROL_DATA_XPTRM;

        XP_N  =1;
        XP_U  =&WORLD_MODEL_CONTROL_DATA_XP;

        THETTRLIMIT_N       =1;
        THETTRLIMIT_LL      =&WORLD_MODEL_CONTROL_DATA_THETTRLL;
        THETTRLIMIT_UL      =&WORLD_MODEL_CONTROL_DATA_THETTRUL;

        SUMJ2_N      =1;
        SUMJ2_SA1    =1;
        SUMJ2_SA2    =1;

        KXPTHR_N     =1;
        KXPTHR_K     =&WORLD_MODEL_CONTROL_DATA_KXPTHR;

        SUMJ4_N      =1;
        SUMJ4_SA1    =1;
        SUMJ4_SA2    =1;

        BIAS_N       =1;
        BIAS_K       =&WORLD_MODEL_CONTROL_DATA_THETTRBIAS;

        THETTROUT_N =1;

        DEG2RAD_N    =1;
        DEG2RAD_K    =&WORLD_DATA_D2R;

GOTO YAWANALOGSAS
        YAWASHOUT_N =1;
        YAWASHOUT_NUM       =&WORLD_MODEL_CONTROL_DATA_YAWOUTNUM;
        YAWASHOUT_DEN       =&WORLD_MODEL_CONTROL_DATA_YAWOUTDEN;

        ZERO_N       =1;
        ZERO_U       =0;

        YAGAIN4_N    =1;
        YAGAIN4_K    =&WORLD_MODEL_CONTROL_DATA_YAGAIN4;

        YASUM2_N     =1;
        YASUM2_SA1   =1;
        YASUM2_SA2   =1;

        VX50KTSWITCH2_N    =1;
        VX50KTSWITCH2_SW   =&WORLD_MODEL_CONTROL_DATA_V50KTS;

        YAGAIN3_N    =1;
        YAGAIN3_K    =&WORLD_MODEL_CONTROL_DATA_YAGAIN3;

        VX50KTSWITCH1_N    =1;
        VX50KTSWITCH1_SW   =&WORLD_MODEL_CONTROL_DATA_V50KTS;

        YALIMIT3_N  =1;
        YALIMIT3_LL =&WORLD_MODEL_CONTROL_DATA_YALLIMIT3;
        YALIMIT3_UL =&WORLD_MODEL_CONTROL_DATA_YAULIMIT3;

        DSAS_N      =1;
        DSAS_SW     =&WORLD_MODEL_CONTROL_DATA_DSAS;

        ASAS_N      =1;
        ASAS_SW     =&WORLD_MODEL_CONTROL_DATA_ASAS;

        YASUM3_N    =1;
        YASUM3_SA1  =1;
        YASUM3_SA2  =-1;

        YAGAIN5_N   =1;
        YAGAIN5_K   =2;

        YAGROUND_N  =1;
        YAGROUND_K  =0;

        YASUM1_N    =1;
        YASUM1_SA1  =1;
        YASUM1_SA2  =1;

        YAGAIN2_N   =1;
        YAGAIN2_K   =&WORLD_MODEL_CONTROL_DATA_YAGAIN2;

        YAGAIN1_N   =1;
        YAGAIN1_K   =&WORLD_MODEL_CONTROL_DATA_YAGAIN1;

        YALAG_N     =1;
        YALAG_NUM   =&WORLD_MODEL_CONTROL_DATA_YALAGNUM;
        YALAG_DEN   =&WORLD_MODEL_CONTROL_DATA_YALAGDEN;

        YALIMIT1_N  =1;
        YALIMIT1_LL =&WORLD_MODEL_CONTROL_DATA_YALLIMIT1;
        YALIMIT1_UL =&WORLD_MODEL_CONTROL_DATA_YAULIMIT1;

        YALIMIT2_N  =1;
        YALIMIT2_LL =&WORLD_MODEL_CONTROL_DATA_YALLIMIT2;
        YALIMIT2_UL =&WORLD_MODEL_CONTROL_DATA_YAULIMIT2;


PARENTG
        SUMJ3_N     =1;
        SUMJ3_SA1   =1;
        SUMJ3_SA2   =1;

        GAIN1_N     =1;
        GAIN1_K     =1.0;

        PRISERVO_N  =2;
        PRISERVO_NUM         =&WORLD_MODEL_CONTROL_DATA_PRISERVONUM;
        PRISERVO_DEN         =&WORLD_MODEL_CONTROL_DATA_PRISERVODEN;


PARENTG


PARENTG


PARENTG


PARENTG
```

```
/////////////////////////////////////////////////////////
//
// file control.prolog
//
// date 31 Mar 1995
// prolog file containing necessary data for SH-60B control
//    system
//
/////////////////////////////////////////////////////////
// useful constants
//
pi=acos(-1);
d2r=pi/180        "degrees to radians conv factor";
r2d=180/pi        "radians to degrees conv factor";
k2f=6076.115/3600 "knots to fps conv factor";
f2k=3600/6076.115 "fps to knots conv factor";
//
// control system data
//
theta1=17.821*d2r "main rotor pitch (rad)";
theta1d=0;
theta1dd=0;
a1s1=-.080*d2r          "lateral cyclic pitch (rad)";
a1s1d=0;
a1s1dd=0;
b1s1=6.391*d2r          "long cyclic pitch (rad)";
b1s1d=0;
b1s1dd=0;
theta2=11.237*d2r "tail rotor pitch (rad)";
phase=-4.0*d2r          "swashplate phase angle";
world_data_pfrasens=zeros(9,1);
world_data_pfrlsens=zeros(9,1);
//
// pilot controls, trim signals
//
Xa=0.0;
Xb=0.0;
Xc=0.0;
Xp=0.0;

world_data_xa=&xa;
world_data_xb=&xb;
world_data_xc=&xc;
world_data_xp=&xp;

Xatrm=4.95;
Xbtrm=1.196;
Xctrm=5.00;
Xptrm=1.925;

a1schk=0.0;
```

```
b1schk=0.0;
th0chk=0.0;
thrchk=0.0;

//
// control system bias inputs
//
A1sbias=-8.0            "lateral cyclic bias (degs)";
B1sbias=9.7754          "longitudinal cyclic bias (degs)";
Thetmrbias=9.4          "main rotor collective bias (degs)";
Thettrbias=21.9         "tail rotor collective bias (degs)";
//
// Cockpit control limits
//
A1sll=0.0         "lateral cyclic stick lower limit";
A1sul=10.0        "lateral cyclic stick upper limit";

B1sll=0.0         "longitudinal cyclic stick lower limit";
B1sul=10.0        "longitudinal cyclic stick upper limit";

Thetmrll=0.0            "main rotor collective lower limit";
Thetmrul=10.0           "main rotor collective upper limit";

Thettrll=0.0            "tail rotor collective lower limit";
Thettrul=5.38           "tail rotor collective upper limit";
//
// stick to swashplate scale factors
//
kxaa1s=1.6        "lateral stick to swashplate scale factor";
kxbb1s=-2.83            "longitudinal to swashplate scale factor";
kxcth0=1.6842           "main  rotor  collective  to  swashplate  scale
factor";
kxpthr=-5.539           "tail  rotro  collective  to  swashplate  scale
factor";
//
// Numerator and denominator for transfer functions for primary servos
//
priservonum=[0 0 1]           "Primary servo tf numerator";
priservoden=[.00114 .0473 1]  "Primary servo tf denominator";
//
// Numerator and denominator for transfer functions for sensors
//
snum=[0 0 1]            "Sensor tf numerator";
sden=[.00014 .016 1]    "Sensor tf denominator";

latacelnum=[0 0 1]            "Lateral accel tf numerator";
latacelden=[.0254 .233 1]     "Lateral accel tf denominator";

//
// Inputs for digital and analog pitch sas
//
```

```
asas=1                      "Analog SAS switch 1 on/0 off";
dsas=1                      "Digital SAS switch 1 on/0 off";


// pitch analog inputs
pawoutnum=[7 0];
pawoutden=[7 1];
pagain1=-.546;
pagain2=-.559;
palagnum=[1 0];
palagden=[1.993 1];
pallimit=-6.0;
paulimit=6.0;
// pitch digital inputs
pdwoutnum=[7 0];
pdwoutden=[7 1];
pdgain1=-.54;
pdgain2=-.55;
pdlagnum=[0 1];
pdlagden=[2 1];
pdllimit=-5.0;
pdulimit=5.0;
// pitch sas input to mechanical controls
psasactnum=[0 0 1];
psasactden=[.00032 .025 1];
psasllimit=-10.0;
psasulimit=10.0;
kpcdgp=-.283;
kxbb1sinv=1/kxbb1s;
//
// Inputs for digital and analog roll sas
//
// roll analog inputs
ragain1=-.103;
ragain2=-.204;
ragain3=-.5;
rallimit1=-29.18;
raulimit1=29.18;
rallimit2=-2.1;
raulimit2=2.1;
ralagnum=[0 1];
ralagden=[2.04 1];
ralag2num=[0 1];
ralag2den=[.02 1];
rallimit3=-5.0;
raulimit3=5.0;
// roll digital inputs
rdgain1=-.103;
rdgain2=-.204;
rdgain3=-.5;
rdllimit1=-2.1;
```

```
rdulimit1=2.1;
rdllimit2=-5.0;
rdulimit2=5.0;
// roll sas input to mechanical controls
rsasactnum=[0 0 1];
rsasactden=[.00032 .025 1];
rsasllimit=-10.0;
rsasulimit=10.0;
kpcdgr=.16;
kxaalsinv=1/kxaals;
//
// Inputs for digital and analog yaw sas
//
// yaw analog inputs
yawoutnum=[2 0];
yawoutden=[2 1];
yallimit1=-9.928;
yaulimit1=9.928;
yallimit2=-6.0;
yaulimit2=6.0;
v50kts=0;
yagain1=-.61;
yagain2=-1.013;
yagain3=-1.229;
yagain4=.152;
yalagnum=[0 1];
yalagden=[2 1];
yallimit3=-6.0;
yaulimit3=6.0;
// yaw digital inputs
ydwoutnum=[2 0];
ydwoutden=[2 1];
ydlag1num=[0 1];
ydlag1den=[1.045 1];
ydlag2num=[0 1];
ydlag2den=[1.045 1];
ydgain1=-.611;
ydgain2=-1.013;
ydgain3=-1.23;
ydgain4=.152;
ydllimit=-5.0;
ydulimit=5.0;
// yaw sas input to mechanical controls
ysasactnum=[0 0 1];
ysasactden=[.00032 .025 1];
ysasllimit=-10.0;
ysasulimit=10.0;
kpcdgy=-.298;
kxpthrinv=1/kxpthr;
//
// Inputs for Pitch Bias Actuator
```

```
//
pbasw=1                  "PBA on/off switch 0-off 1-on";
pbagain1=-.375;
pbagain2=-.13125;
pbagain3=10.0;
pbathetllimit=-20.0;
pbathetulimit=20.0;
pbaqllimit=-34.5;
pbaqulimit=34.5;
pbabias=-4.9;
pballimit1=-3.0;
pbaulimit1=3.0;
pballimit2=-15.0;
pbaulimit2=15.0;

pbamap=[0; 0; 0; 0; 0; 0; 0; 0; 0; .525; 1.050; 1.575; 2.1;..
2.625; 3.150; 3.675; 4.2; 4.725; 5.25; 5.25; 5.25]';

xcs=0.0;

gainmap=[0; 0; 0; 0; -.4333; -.08667; -1.3; -1.3; -1.3; -1.3;..
 -1.3; -1.3;-1.3; -1.3; -1.3; -1.3;-1.3; -1.3; -1.3]';
path("$FL_LOCAL_DB/tables/rotorcraft")
load("stab.sav");

// end of control.prolog
```

```
//////////////////////////////////////////////////////////
// file control.configure
//
// date 4 April 1995
//////////////////////////////////////////////////////////

popg
pushg(data)
  v50kts=&world_model_control_flightcontrols_directional_ ..
         yawdigitalsas_yawswitch_y;
popg
pushg(cpg)
// Control system input and outputs

   xatrm= &world_model_control_data_xatrm ..
      "Lateral stick trim position (inches)";

   xbtrm= &world_model_control_data_xbtrm ..
      "Longitudinal stick trim position (inches)";

   xctrm= &world_model_control_data_xctrm ..
      "Collective stick trim position (inches)";

   xptrm= &world_model_control_data_xptrm ..
      "Pedal trim position (inches)";
//
// Following commands allow cockpit controls and SAS system
//  to be accessed in Initialization/Configuration/Flight Control
//  in Model Analysis
//
pushg(par)

   xatrm= &world_model_control_data_xatrm ..
      "Lateral stick trim position (inches)";

   xbtrm= &world_model_control_data_xbtrm ..
      "Longitudinal stick trim position (inches)";

   xctrm= &world_model_control_data_xctrm ..
      "Collective stick trim position (inches)";

   xptrm= &world_model_control_data_xptrm ..
      "Pedal trim position (inches)";

   asas=&world_model_control_data_asas ..
      "Analog SAS switch (0-off / 1-on)";

   dsas=&world_model_control_data_dsas ..
      "Digital SAS switch (0-off / 1-on)";

   pbasw=&world_model_control_data_pbasw ..
```

```
        "Pitch Bias Actuator Switch (0-off / 1-on)";


popg
//
// Following commands allow cockpit control inputs to be
//   accessed as inputs into the model in model analysis.
//
pushg(in)

   xa= &world_model_control_data_xa ..
      "Lateral cyclic stick input (inches)";

   xb= &world_model_control_data_xb ..
      "Longitudinal cyclic stick input (inches)";

   xc= &world_model_control_data_xc ..
      "Collective stick input (inches)";

   xp= &world_model_control_data_xp ..
      "Pedal input (inches)";


popg

   b1s   = &world_model_control_flightcontrols_longitudinal_b1sout_y ..
          "Longitudinal swashplate angle (rad)";

   a1s   = &world_model_control_flightcontrols_lateral_a1sout_y ..
            "Lateral swashplate angle (rad)";

   mth   = &world_model_control_flightcontrols_collective_theta0out_y ..
               "Main rotor collective swashplate angle (rad)";

   tth   = &world_model_control_flightcontrols_directional_thettrout_y ..
              "Tail rotor collective swashplate angle (rad)";

   a1schk = &world_model_control_data_a1schk;

   b1schk = &world_model_control_data_b1schk;

   th0chk = &world_model_control_data_th0chk;

   thrchk = &world_model_control_data_thrchk;

   xa       = &world_model_control_data_xa ..
      "Pilot's lateral stick input (inches)";

   xb       = &world_model_control_data_xb ..
      "Pilot's longitudinal stick input (inches)";
   xc      = &world_model_control_data_xc ..
      "Pilot's main rotor collective stick input (inches)";

   xp      = &world_model_control_data_xp ..
      "Pilot's pedal input (inches)";

   xbtrmy  =&world_model_control_flightcontrols_longitudinal_xbtrm_y;
   xatrmy  =&world_model_control_flightcontrols_lateral_xatrm_y;
   xctrmy  =&world_model_control_flightcontrols_collective_xctrm_y;
   xptrmy  =&world_model_control_flightcontrols_directional_xptrm_y;

   xby     =&world_model_control_flightcontrols_longitudinal_xb_y;
   xay     =&world_model_control_flightcontrols_lateral_xa_y;
   xcy     =&world_model_control_flightcontrols_collective_xc_y;
   xpy     =&world_model_control_flightcontrols_directional_xp_y;

   world_model_control_flightcontrols_longitudinal_pba_pba3mp_var = ..
      &world_model_control_sensors_uknots_y;

   world_model_control_stabilator_gainmap_var = ..
        &world_model_control_stabilator_vxlag_y;

   world_model_control_stabilator_stabpositmap_var = ..
        &world_model_control_stabilator_stabinput_y;

varlist @list + ..
   world_model_control_cpg_xbtrm, world_model_control_cpg_xatrm, ..
   world_model_control_cpg_xctrm, world_model_control_cpg_xptrm;

varlist @uprtrb + ..
   xb,xc,xp,xp,b1schk,a1schk,th0chk,thrchk;


//
// End of cont.configure
```

```
//////////////////////////////////////////////////////////
// file control.epilog
// date 4 April 1995
// Epilog file for SH-60B control system
// Connections between control system creted in Gscope and
//  remainder of model created under Model Tree are executed
//  here.
//////////////////////////////////////////////////////////
//
world_data_a1s1=&world_model_control_flightcontrols_lateral_a1sout_y;
world_data_b1s1=&world_model_control_flightcontrols_longitudinal_b1sout_y;
world_data_theta1=&world_model_control_flightcontrols_collective_theta0o
ut_y;
world_data_theta2=&world_model_control_flightcontrols_directional_thettr
out_y;
world_data_alfiv=&world_model_control_sensors_alpha_y;
world_data_betiv=&world_model_control_sensors_beta_y;
world_data_alfwf=&world_model_airframe_wf_alpha;
world_data_betwf=&world_model_airframe_wf_beta;

control_sensors_dofsensora_u=&world_data_pfrasens;
control_sensors_dofsensorl_u=&world_data_pfrlsens;

world_model_airframe_hstab1_eitail_u = ..
        &world_model_control_stabilator_stabposit_y

world_model_airframe_hstab2_eitail_u = ..
        &world_model_control_stabilator_stabposit_y

exec("controlslv.exc",1)

//end of control.epilog
```

APPENDIX D. EXECUTABLE SCRIPT FILES

```
/////////////////////////////////////////////////////////////////////////
// file trimsweep.scr
// 21 July 1995
//Script to obtain lateral/longitudinal cyclic stick, collective,
// pedal position and stabilator incidence angle for airspeed from
// 0 to 150 knots in 30 knot increments.
//
// Output utrim has as its columns longitudinal, lateral, collective,
//    pedal, roll attitide and pitch attitude.
//    The rows correspond to airspeed.
/////////////////////////////////////////////////////////////////////////
cpg_trimg0=.5     // sets step size for trim routine
//
cpg_trimiter=40   // sets maximum number of iterations.  If trim is not
//               accomplished by value set, script stops
utrim=[];
stab=[];
i=1;
for ss=0:30:150
      world_cpg_veq=ss
      exec("Compute_Body_Axis_Velocity.exc",1)
      trim(0)
      sp=world_model_control_stabilator_stabposit_y(1);
      stab(i)=sp;
      utrim=[utrim ; @utrim'];
      i=i+1;
end
//convert pitch and roll angle to degrees
utrim(:,5)=57.2958*utrim(:,5);
utrim(:,6)=57.2958*utrim(:,6);
//convert stab incidence angle to degrees
stab=57.2958*stab';
```

```
////////////////////////////////////////////////////////////////////////////
// file power_req.scr
// 10 August 1995
// Script to obtain main rotor power and total power required (i.e. main
// rotor plus tail rotor power) for a user specified airspeed range.
//   THIS IS ONLY POWER REQUIRED BY MAIN AND TAIL ROTORS TO MEET
//    SPECIFIED WEIGHT AND AIRSPEED, NOT POWER OUTPUT OF ENGINES.
////////////////////////////////////////////////////////////////////////////
//
cpg_trimg0=.5     // sets step size for trim routine
//
cpg_trimiter=40   // sets maximum number of iterations.  If trim is not
//              accomplished by value set, script stops
// User enters aircraft weight in pounds.
wt=enter("Enter aircraft weight in lbs.");
world_model_airframe_data_fmass=wt/32.2;
//
// User enter equivalent airspeed and step size in knots.
vlo=enter("Enter beginning EAS in kts.")
vhi=enter("Enter ending EAS in kts.")
inc=enter("Enter step size.")
i=1;
for ss=vlo:inc:vhi
      world_cpg_veq=ss
      exec("Compute_Body_Axis_Velocity.exc",1)
      trim(0)
      theta=world_results_ut(6);
      mrtorquex=abs(world_model_rotor1_rotor_rotorhead_hub_pfo(4));
      mrtorquez=abs(world_model_rotor1_rotor_rotorhead_hub_pfo(6));
      mrtorque=abs(mrtorquex*sin(theta))+abs(mrtorquez*cos(theta));
      mromega=world_model_rotor1_rotor_data_rpmnom;
      trtorque=abs(world_model_rotor2_rotor_trotor_q);
      tromega=world_model_rotor2_rotor_data_rpmnom;
      power=(mrtorque*mromega/550.0)+(trtorque*tromega/550.0);
      MRHP(i)=mrtorque*mromega/550.0;
      HPTOTAL(i)=power;
      VEL(i)=ss;
      i=i+1;
end
HPTOTAL=HPTOTAL';
VEL=VEL';
MRHP=MRHP';
```

```
//Linear Frequency Response
function a=LinearFrequencyResponse(b)
//This function generates a frequency response from the last linearized
model
//
function cfreq;
a=0;    //dummy return variable
s=world_results_s;
[srow,scol]=size(s);
ns=world_results_ns;
nptfft=2048;
// dfrq=2*world_cpg_pi/(nptfft*world_data_dt);
ww=logspace(.1,100,500);
//Break up the s matrix
f=s(1:ns,1:ns);
g=s(1:ns,ns+1:scol);
h=s(ns+1:srow,1:ns);
d=s(ns+1:srow,ns+1:scol);
[amp,phase] = cfreq(f,g,h,d,ww);
//[amp,phase]=freq(s,ns,ww);
world_results_ww=ww;
world_results_amplin=amp;
world_results_phaselin=phase*world_data_r2d;
end   //LinearFrequencyResponse
```

APPENDIX E. OPEN LOOP EIGENANALYSIS

$$\dot{x}=Ax+Bu$$

$x=[u;w;q;\theta;v;p;\phi;r]$
$u=[\delta_e;\delta_a;\delta_c;\delta_p]$

```
Equivalent Airspeed = 0 kts, Gross Weight = 19462 lbs, SSL Conditions
A=
[0.2032   0.0140   1.1439 -32.1410 -0.2340 -1.7236 -0.0001 -0.0341;
 0.2417  -0.2776  -0.0737  -1.9453 -0.2371 -0.4255  1.4656  2.7489;
 0.0623   0.0009  -0.5504  -0.0001 -0.0594  0.1383  0.0000  0.0091;
 0.0000   0.0000   0.9990   0.0000  0.0000  0.0000  0.0000  0.0456;
-0.0009  -0.0008  -1.4501   0.0892 -0.0535 -1.3502 32.1074  0.4311;
 0.0015  -0.0005  -1.3684   0.0002 -0.0299 -2.7869  0.0000  0.2011;
 0.0000   0.0000  -0.0028   0.0000  0.0000  1.0000  0.0000  0.0607;
-0.0001  -0.0007  -0.0115  -0.0001  0.0070  0.1149  0.0000 -0.2346];

B=
  [-1.6181    -0.0037     0.7799     0.9390;
    0.0065     0.0145    -7.7573     0.3274;
    0.3310     0.0097    -0.0302    -0.0536;
    0.0000     0.0000     0.0000     0.0000;
    0.0206     0.9487    -0.0059    -0.9555;
    0.0876     1.1726    -0.2588    -0.6348;
    0.0000     0.0000     0.0000     0.0000;
   -0.0029    -0.0344     0.1191     0.3583];

>> [eigvec,eigval]=eig(A)
eigvec =

  Columns 1 through 4
  -0.1704        0.8566          0.5466 - 0.7038i   0.5466 + 0.7038i
  -0.1559        0.0321          0.0085 - 0.1412i   0.0085 + 0.1412i
  -0.0060       -0.0388         -0.0047 - 0.0233i  -0.0047 + 0.0233i
   0.0018        0.0318         -0.0245 - 0.0079i  -0.0245 + 0.0079i
  -0.9437        0.5116          0.3418 - 0.2600i   0.3418 + 0.2600i
  -0.2242        0.0242          0.0020 + 0.0113i   0.0020 - 0.0113i
   0.0755       -0.0194          0.0118 + 0.0040i   0.0118 - 0.0040i
   0.0118       -0.0067          0.0015 - 0.0019i   0.0015 + 0.0019i

  Columns 5 through 8

   0.6594 + 0.3367i   0.6594 - 0.3367i   0.4681              0.0535
   0.1001 + 0.0422i   0.1001 - 0.0422i   0.7163              0.9958
   0.0108 + 0.0064i   0.0108 - 0.0064i  -0.0020             -0.0005
   0.0083 - 0.0143i   0.0083 + 0.0143i   0.0018              0.0007
   0.6450 + 0.1517i   0.6450 - 0.1517i   0.5161              0.0739
  -0.0121 - 0.0015i  -0.0121 + 0.0015i  -0.0019             -0.0002
  -0.0024 + 0.0158i  -0.0024 - 0.0158i  -0.0023             -0.0006
  0.0020 - 0.0031i   0.0020 + 0.0031i   0.0370             0.0067



eigval=
  Columns 1 through 4

 -2.9590                  0                  0                  0
       0            -1.2284                  0                  0
       0                  0        0.4479 + 0.8051i             0
       0                  0                  0        0.4479 - 0.8051i
       0                  0                  0                  0
       0                  0                  0                  0
       0                  0                  0                  0
       0                  0                  0                  0

  Columns 5 through 8

       0                  0                  0                  0
       0                  0                  0                  0
       0                  0                  0                  0
       0                  0                  0                  0
  0.0073 + 0.7587i        0                  0                  0
       0        0.0073 - 0.7587i             0                  0
       0                  0            -0.1569                  0
       0                  0                  0            -0.2659
```

```
Equivalent Airspeed = 30 kts, Gross Weight = 19462 lbs, SSL Conditions
A=
[-0.0479  0.0008 -0.9292 -32.1745 -0.0061 -1.4327  0.0001  -0.3472;
 -0.1812 -0.4337 51.0366  -1.2781 -0.0150 -0.8065  0.8621   2.0679;
 -0.0061  0.0028 -0.6877   0.0000  0.0053  0.2174 -0.0001  -0.0664;
  0.0000  0.0000  0.9996   0.0000  0.0000  0.0000  0.0000   0.0268;
  0.0127  0.0113 -1.3448   0.0344 -0.0750  0.1587 32.1609 -49.5333;
  0.0142  0.0283 -1.3644   0.0001 -0.0433 -2.9563 -0.0013   0.6619;
  0.0000  0.0000 -0.0011   0.0000  0.0000  1.0000  0.0000   0.0398;
 -0.0056 -0.0042  0.0383   0.0000  0.0112  0.0423 -0.0001  -0.4823];

B=
  [-1.5424   -0.0037    0.7102    0.8955;
    -1.2456    0.1050   -7.0568    1.0599;
     0.3424    0.0064    0.0370   -0.0559;
     0.0000    0.0000    0.0000    0.0000;
     0.0231    0.9466    0.0522   -1.0234;
     0.1124    1.1726   -0.0939   -0.6949;
     0.0000    0.0000    0.0000    0.0000;
    -0.0144   -0.0371    0.0733    0.3904];
>> [eigvec,eigval]=eig(A)

eigvec =

  Columns 1 through 4

  -0.1943          0.2759             0.0086 - 0.0788i   0.0086 + 0.0788i
  -0.6130          0.8789            -0.0541 - 0.0730i  -0.0541 + 0.0730i
   0.0234         -0.0119             0.0010 - 0.0015i   0.0010 + 0.0015i
  -0.0082          0.0096            -0.0020 - 0.0007i  -0.0020 + 0.0007i
  -0.7312          0.3882             0.8445 - 0.5213i   0.8445 + 0.5213i
  -0.2136          0.0166            -0.0114 + 0.0090i  -0.0114 - 0.0090i
   0.0742         -0.0136             0.0109 + 0.0103i   0.0109 - 0.0103i
   0.0053          0.0009            -0.0015 - 0.0103i  -0.0015 + 0.0103i

  Columns 5 through 8

  -0.8642              0.4499 - 0.4691i   0.4499 + 0.4691i   0.3162
   0.4851              0.5022 - 0.0151i   0.5022 + 0.0151i  -0.0781
   0.0053              0.0033 + 0.0015i   0.0033 - 0.0015i  -0.0009
   0.0126             -0.0021 - 0.0080i  -0.0021 + 0.0080i   0.0009
  -0.1327              0.5229 - 0.2273i   0.5229 + 0.2273i   0.9443
   0.0003             -0.0019 - 0.0001i  -0.0019 + 0.0001i  -0.0070
   0.0009              0.0023 + 0.0036i   0.0023 - 0.0036i   0.0389
   0.0017              0.0021 - 0.0024i   0.0021 + 0.0024i   0.0268
```

```
eigval =

  Columns 1 through 4

 -2.8769              0                     0                     0
       0        -1.2267                     0                     0
       0              0      -0.1584 + 0.9440i                    0
       0              0                     0     -0.1584 - 0.9440i
       0              0                     0                     0
       0              0                     0                     0
       0              0                     0                     0
       0              0                     0                     0

  Columns 5 through 8

       0                   0                     0              0
       0                   0                     0              0
       0                   0                     0              0
       0                   0                     0              0
  0.4249                   0                     0              0
       0   -0.2667 + 0.3470i                     0              0
       0                   0     -0.2667 - 0.3470i              0
       0                   0                     0        -0.1539
```

Equivalent Airspeed = 60kts, Gross Weight = 19462 lbs, SSL Conditions

```
A=
[-0.0273  0.0111  -0.3895 -32.1942 -0.0009 -1.3126  0.7351   -0.2727;
 -0.0627 -0.6118 100.2964  -0.5964 -0.0182 -1.6127  2.7406    2.0191;
  0.0040  0.0051  -0.9507   0.0003  0.0024  0.1547  0.7213   -0.0872;
  0.0000  0.0000   0.9998   0.0000  0.0000  0.0000  0.0000    0.0212;
  0.0023  0.0064  -1.0570   0.0127 -0.0886 -0.2243 32.1609 -100.2527;
  0.0008  0.0254  -1.1231   0.0000 -0.0496 -3.0257 -0.0055    0.6421;
  0.0000  0.0000  -0.0004   0.0000  0.0000  1.0000  0.0000    0.0186;
 -0.0028 -0.0077   0.0236   0.0000  0.0129  0.0574  0.0001   -0.5019];

B=
  [-1.5485    0.0143    0.4059    0.8824;
    -2.8819    0.2766   -7.9040    1.9588;
     0.3630    0.0237    0.0751   -0.0800;
     0.0000    0.0000    0.0000    0.0000;
     0.0391    0.9314    0.1764   -1.1678;
     0.1753    1.1593    0.1036   -0.8188;
     0.0000    0.0000    0.0000    0.0000;
    -0.0310   -0.0360    0.0042    0.4500];

>> [eigvec,eigval]=eig(A)

eigvec =

  Columns 1 through 4

  -0.1584            0.1168              0.1152 - 0.0132i   0.1152 + 0.0132i
   0.0248            0.9220              0.3301 - 0.3066i   0.3301 + 0.3066i
  -0.0108           -0.0064              0.0046 + 0.0030i   0.0046 - 0.0030i
   0.0033            0.0047              0.0012 - 0.0041i   0.0012 + 0.0041i
  -0.8971            0.3688              0.7276 + 0.5036i   0.7276 - 0.5036i
  -0.3925            0.0085             -0.0142 - 0.0079i  -0.0142 + 0.0079i
   0.1230           -0.0063             -0.0029 + 0.0121i  -0.0029 - 0.0121i
   0.0127            0.0027              0.0073 - 0.0038i   0.0073 + 0.0038i

  Columns 5 through 8

   0.5425 - 0.3307i   0.5425 + 0.3307i  -0.6051             0.9631
   0.0309 - 0.6479i   0.0309 + 0.6479i   0.7349            -0.2666
   0.0039 - 0.0047i   0.0039 + 0.0047i   0.0018            -0.0015
  -0.0073 - 0.0082i  -0.0073 + 0.0082i  -0.0055            -0.0072
   0.0501 - 0.4157i   0.0501 + 0.4157i   0.3061            -0.0348
  -0.0013 + 0.0035i  -0.0013 - 0.0035i   0.0004            -0.0011
   0.0060 + 0.0033i   0.0060 - 0.0033i  -0.0014            -0.0056
  -0.0005 + 0.0015i  -0.0005 - 0.0015i   0.0003            -0.0017
```

```
eigval =

  Columns 1 through 4

 -3.1903                  0                        0                        0
       0            -1.3536                        0                        0
       0                  0          -0.3578 + 1.2448i                      0
       0                  0                        0          -0.3578 - 1.2448i
       0                  0                        0                        0
       0                  0                        0                        0
       0                  0                        0                        0
       0                  0                        0                        0

  Columns 5 through 8

       0                  0                        0                        0
       0                  0                        0                        0
       0                  0                        0                        0
       0                  0                        0                        0
  0.0860 + 0.5440i        0                        0                        0
       0            0.0860 - 0.5440i               0                        0
       0                  0                  -0.3265                        0
       0                  0                        0                   0.2078
```

Equivalent Airspeed = 90 kts, Gross Weight = 19462 lbs, SSL Conditions

```
A=
[-0.0382  0.0264   1.6689 -32.2000 -0.0024 -1.1561  0.5059   -0.2737;
-0.0149 -0.6866 149.8034   0.0205 -0.0191 -2.5297  3.5837    2.2428;
 0.0007  0.0094  -1.1287   0.0000  0.0006  0.1460  1.0649   -0.0805;
 0.0000  0.0000   0.9997   0.0000  0.0000  0.0000  0.0000    0.0236;
 0.0035  0.0002  -0.9080  -0.0004 -0.1145 -2.1136 31.9384 -150.8165;
 0.0026  0.0221  -1.0104   0.0000 -0.0556 -2.9219 -0.0903    0.7450;
 0.0000  0.0000   0.0000   0.0000  0.0000  1.0000  0.0000   -0.0006;
-0.0022-0.0081    0.0189   0.0000  0.0157  0.0579  0.0006   -0.5714];

B=
   [0.3274   -0.0001   -1.3174    0.7591;
   -8.9522    0.4225   -4.5277    2.9016;
    0.0997    0.0316    0.3734   -0.0805;
    0.0000    0.0000    0.0000    0.0000;
    0.1997    0.9315   -0.0186   -1.3339;
    0.1588    1.1544    0.1728   -0.9480;
    0.0000    0.0000    0.0000    0.0000;
   -0.0259   -0.0344   -0.0385    0.5281]

>>     [eigvec,eigval]=eig(A)

eigvec =

  Columns 1 through 4

   0.0124             0.0571 - 0.0091i   0.0571 + 0.0091i  -0.0694
  -0.9264             0.3469 - 0.2908i   0.3469 + 0.2908i  -0.9472
   0.0206             0.0035 + 0.0029i   0.0035 - 0.0029i   0.0065
  -0.0063             0.0011 - 0.0026i   0.0011 + 0.0026i  -0.0036
   0.3159             0.6254 + 0.6325i   0.6254 - 0.6325i  -0.3127
   0.1941            -0.0157 - 0.0090i  -0.0157 + 0.0090i  -0.0106
  -0.0600            -0.0029 + 0.0105i  -0.0029 - 0.0105i   0.0060
  -0.0090             0.0076 - 0.0031i   0.0076 + 0.0031i  -0.0020


  Columns 5 through 8

  -0.0845 - 0.5171i  -0.0845 + 0.5171i   0.9167             0.9090
  -0.7378 - 0.1899i  -0.7378 + 0.1899i  -0.3944            -0.4104
  -0.0037 - 0.0036i  -0.0037 + 0.0036i  -0.0024            -0.0012
  -0.0084 + 0.0043i  -0.0084 - 0.0043i  -0.0091             0.0052
  -0.3647 - 0.1087i  -0.3647 + 0.1087i  -0.0640            -0.0719
   0.0028 + 0.0011i   0.0028 - 0.0011i  -0.0001            -0.0005
   0.0036 - 0.0042i   0.0036 + 0.0042i  -0.0002             0.0021
   0.0011 + 0.0006i   0.0011 - 0.0006i   0.0001             0.0004
```

```
eigval =

  Columns 1 through 4

  -3.2319                  0                        0                        0
        0            -0.4240 + 1.6105i              0                        0
        0                  0                  -0.4240 - 1.6105i              0
        0                  0                        0                  -1.7634
        0                  0                        0                        0
        0                  0                        0                        0
        0                  0                        0                        0
        0                  0                        0                        0


  Columns 5 through 8

        0                  0                        0                        0
        0                  0                        0                        0
        0                  0                        0                        0
        0                  0                        0                        0
   0.1753 + 0.5095i        0                        0                        0
        0             0.1753 - 0.5095i              0                        0
        0                  0                   0.2656                        0
        0                  0                        0                  -0.2342
```

```
Equivalent Airspeed = 120kts, Gross Weight = 19462 lbs, SSL Conditions
A=
[-0.0512  0.0252  13.1020 -32.1476 -0.0023  -1.1795  0.4129    0.1094;
 -0.0016 -0.7566 198.9821   1.8406 -0.0343  -4.0876  5.9117    2.7723;
  0.0022  0.0079  -1.3845  -0.0033 -0.0028   0.1522  1.7174   -0.0599;
  0.0000  0.0000   0.9994   0.0000  0.0000   0.0000  0.0000    0.0337;
  0.0057 -0.0050  -0.9125  -0.0622 -0.1424 -13.4599 32.0630 -200.9072;
  0.0055  0.0216  -1.0326   0.0000 -0.0683  -2.7792  0.0046    0.9592;
  0.0000  0.0000   0.0019   0.0000  0.0000   1.0000  0.0000   -0.0575;
 -0.0024 -0.0057  -0.0452   0.0000  0.0182   0.0623  0.0021   -0.6740];

B=
  [-1.4526   -0.0207    0.1618    0.8386;
    -6.7201   0.4973   -9.2217    4.0627;
     0.4580   0.0539    0.1217   -0.1434;
     0.0000   0.0000    0.0000    0.0000;
    -0.1309   0.9671    0.2090   -1.4605;
     0.1745   1.1755    0.2126   -1.0776;
     0.0000   0.0000    0.0000    0.0000;
    -0.0261  -0.0351   -0.0181    0.5859];

>> [eigvec,eigval]=eig(A)

eigvec =

  Columns 1 through 4

  -0.0648             -0.0744 + 0.0217i  -0.0744 - 0.0217i   0.0709
  -0.9677             -0.5893 + 0.2778i  -0.5893 - 0.2778i   0.9653
   0.0159             -0.0033 - 0.0046i  -0.0033 + 0.0046i  -0.0036
  -0.0044             -0.0019 + 0.0024i  -0.0019 - 0.0024i   0.0023
   0.2293             -0.4672 - 0.5922i  -0.4672 + 0.5922i   0.2510
   0.0772              0.0151 + 0.0106i   0.0151 - 0.0106i   0.0068
  -0.0220              0.0036 - 0.0093i   0.0036 + 0.0093i  -0.0044
  -0.0049             -0.0065 + 0.0012i  -0.0065 - 0.0012i   0.0006

 Columns 5 through 8

  -0.3758 + 0.0134i  -0.3758 - 0.0134i   0.9021              0.7165
  -0.6208 + 0.6098i  -0.6208 - 0.6098i  -0.4308             -0.6870
  -0.0045 + 0.0009i  -0.0045 - 0.0009i  -0.0020             -0.0016
  -0.0001 + 0.0076i  -0.0001 - 0.0076i  -0.0089              0.0049
  -0.2319 + 0.2179i  -0.2319 - 0.2179i  -0.0233             -0.1210
   0.0017 - 0.0012i   0.0017 + 0.0012i  -0.0002             -0.0003
  -0.0015 - 0.0031i  -0.0015 + 0.0031i  -0.0011              0.0011
   0.0006 - 0.0000i   0.0006 + 0.0000i  -0.0001              0.0001
```

```
eigval =

  Columns 1 through 4

  -3.5311                  0                        0                        0
        0         -0.4287 + 1.8324i                 0                        0
        0                  0               -0.4287 - 1.8324i                 0
        0                  0                        0                  -1.5508
        0                  0                        0                        0
        0                  0                        0                        0
        0                  0                        0                        0
        0                  0                        0                        0


  Columns 5 through 8

        0                  0                        0                        0
        0                  0                        0                        0
        0                  0                        0                        0
        0                  0                        0                        0
   0.1247 + 0.5880i        0                        0                        0
        0          0.1247 - 0.5880i                 0                        0
        0                  0                   0.2255                        0
        0                  0                        0                  -0.3236
```

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Library, Code 52 2
   Naval Postgraduate School
   Monterey, California 93943-5101

3. Chairman, Department of Aeronautics and Astronautics AA/Co 1
   Naval Postgraduate School
   Monterey, California 93943-5002

4. Dr. E. R. Wood AA/Wd. 6
   Naval Postgraduate School
   Monterey, California 93943-5002

5. Mr. Ron Duval 3
   Advanced Rotorcraft Technologies
   1685 Plymouth St. Suite 250
   Mountain View, California 94043

6. Tom Lawrence 3
   Sikorsky Aircraft
   MS S336A1
   6900 Main St.
   Stratford, Connecticut 06497-9129

7. Dean Carico 3
   MS1 Hangar 110
   NAWCAD
   Patuxent River, Maryland 20670-5304

8. Alan W. Schwartz 3
   Sea-Based Aviation Office, Code 5120
   Bethesda, Maryland 20084-5000

9. Dr. Daniel Schrage 1
   School of Aerospace Engineering
   Georgia Institute of Technology
   Atlanta, Georgia 30332

10. Roy C. Wagner 2
6283 Venture Court
Slatington, Pennsylvania 18080